NAVAL POSTGRADUATE SCHOOL
Monterey, California

THESIS

DESIGN AND INTEGRATION OF A FLIGHT MANAGEMENT SYSTEM FOR THE UNMANNED AIR VEHICLE FROG

by

Timothy C. Rivers

December 1998

Thesis Advisor: Isaac I. Kaminer

Approved for public release; distribution is unlimited.

REPORT DOCUMENTATION PAGE

Form Approved OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	December 1998	Engineer's Thesis

4. TITLE AND SUBTITLE
DESIGN AND INTEGRATION OF A FLIGHT MANAGEMENT SYSTEM FOR THE UNMANNED AIR VEHICLE FROG

5. FUNDING NUMBERS

6. AUTHOR(S)
Rivers, Timothy C.

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Naval Postgraduate School
Monterey, CA 93943-5000

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)

10. SPONSORING / MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES
The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

12a. DISTRIBUTION / AVAILABILITY STATEMENT
Approved for public release; distribution is unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT *(maximum 200 words)*

The purpose of this thesis is to design, integrate and flight test a Flight Management System (FMS) for the computer control of an unmanned air vehicle (UAV). By combining modern control design techniques and the capabilities of a Rapid Prototyping System (RPS), we were able to safely go from concept to flight test in a relatively short amount of time without sacrificing thoroughness in computer simulation, code validation and verification, or hardware-in-the-loop ground testing. This ability to quickly field new or modified flight control systems for UAV's is of ever increasing importance as Department of Defense places greater emphasis on the use of UAV's in widely varying mission areas.

The primary focus of this thesis is on the design and testing of a heading controller. However, to fully integrate this into the FMS, the research and testing includes airspeed and altitude controllers designed by previous thesis students. Also included as part of the implementation process, is a thorough sensor evaluation to ensure the controller inputs are adequate to support the FMS.

The design and test equipment include a highly modified FROG UAV from the U.S. Army, the MATRIX$_X$ Product Family of software tools developed by Integrated Systems, Inc., and a Ground Station built at NPS from commercially available computer and communication equipment.

14. SUBJECT TERMS	15. NUMBER OF PAGES
Unmanned Aerial Vehicles, Rapid Prototyping System, Flight Management System, Heading Controller	111
	16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFI- CATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	UL

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. 239-18

Approved for public release; distribution is unlimited

DESIGN AND INTEGRATION OF A FLIGHT MANAGEMENT SYSTEM FOR THE UNMANNED AIR VEHICLE FROG

Timothy C. Rivers
Commander, United States Navy
B.S., State University of New York at Buffalo, 1975

Submitted in partial fulfillment of the
requirements for the degree of

AERONAUTICAL AND ASTRONAUTICAL ENGINEER

from the

NAVAL POSTGRADUATE SCHOOL
December 1998

Author: Timothy C. Rivers
Timothy C. Rivers

Approved by: I Kaminer
Isaac I. Kaminer, Thesis Advisor

Richard M. Howard
Richard M. Howard, Second Reader

G H Lindsey
Gerald H. Lindsey, Chairman
Department of Aeronautics and Astronautics

ABSTRACT

The purpose of this thesis is to design, integrate and flight test a Flight Management System (FMS) for the computer control of an unmanned air vehicle (UAV). By combining modern control design techniques and the capabilities of a Rapid Prototyping System (RPS), we were able to safely go from concept to flight test in a relatively short amount of time without sacrificing thoroughness in computer simulation, code validation and verification, or hardware-in-the-loop ground testing. This ability to quickly field new or modified flight control systems for UAV's is of ever increasing importance as the Department of Defense places greater emphasis on the use of UAV's in widely varying mission areas.

The primary focus of this thesis is on the design and testing of a heading controller. However, to fully integrate this into the FMS, the research and testing includes airspeed and altitude controllers designed by previous thesis students. Also included as part of the implementation process, is a thorough sensor evaluation to ensure the controller inputs are adequate to support the FMS.

The design and test equipment include a highly modified FROG UAV from the U.S. Army, the MATRIX$_X$ Product Family of software tools developed by Integrated Systems, Inc., and a Ground Station built at NPS from commercially available computer and communication equipment.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS, ACRONYMS AND ABBREVIATIONS

β	Sideslip or yaw angle
γ	Flight path angle
ζ	Damping ratio
ω_n	Natural frequency
H_2O	Water
M_P	Maximum overshoot
t_r	Rise time
A/D	Analog-to-digital
ARI	Aileron-rudder interconnect
BW	Bandwidth
CL	Closed loop
cm	Centimeter
dB	Decibels
deg/sec	Degrees per second
DGPS	Differential Global Positioning System
FMS	Flight management system
fpm	Feet per minute
fps	Feet per second
ft	Feet
FTP	File transfer protocol
Gm	Gain Margin
GPS	Global Positioning System
GUI	Graphical User Interface
HCE	Hardware Connection Editor
Hdg	Heading
I/O	Input/Output
IA	Interactive Animation
IMU	Inertial Measuring Unit
ISI	Integrated Systems, Inc. of Sunnyvale, CA
NPS	Naval Postgraduate School
OL	Open loop
PI	Proportional-plus-Integral
PID	Proportional-plus-Integral-plus-Derivative
Pm	Phase Margin
psi	Pounds per square inch
PW	Pulse width
PWM	Pulse width modulation
rad	Radian
RC	Radio control

RF	Radio frequency
RPS	Rapid prototyping system
UAV	Unmanned air vehicle
UCAV	Unmanned combat air vehicles
VSI	Vertical speed indicator

ACKNOWLEDGMENT

I would like to thank my wife, Joanne, who accepted my absence and carried on preparing for our impending move with unselfish devotion and the fine English tradition that she has brought to our marriage. To my children, who are too young to understand why Daddy couldn't tuck them in at night, I can only offer the promise of better days.

I would like to thank my advisor, Dr. Isaac Kaminer, for the guidance, help, and above all the freedom to discover on my own. The lessons that are learned on your own are remembered the longest.

To the Avionics students who have paved the way before me in designing the FROG control systems and who's work I have referenced, I owe a debt of gratitude for giving me the strong foundation on which to continue their work.

I would also like to give thanks and credit to the following individuals, without whom I could not have gotten this far:

Don Meeks - Highly skilled and experienced R/C model pilot and ingenious inventor of the Beta vane calibrator shown in Appendix A.

Jerry Lentz - Computer/Electronics wizard who single-handedly built the Ground Station that is critical to Flight Management System application.

Mark Watson - Fellow UAV Thesis student who helped with the coding, implementation and flight testing in his "spare time".

I. INTRODUCTION

The purpose of this thesis is to design, integrate and flight test a flight management system (FMS) for the computer control of an unmanned air vehicle (UAV). By combining modern control design techniques and the capabilities of a rapid prototyping system (RPS), we were able to go safely from concept to flight test in a relatively short amount of time. More importantly, it was accomplished without sacrificing thoroughness in computer simulation, code validation and verification, or hardware-in-the-loop ground testing. This ability to quickly field new or modified flight control systems for UAV's is of ever increasing importance as Department of Defense places greater emphasis on the use of UAV's and unmanned combat air vehicles (UCAV) in widely varying mission areas [Ref. 1].

The focus of this thesis is twofold:

1. Evaluate the sensors available for the use by the FMS.
2. Design, test and implement a heading controller.

The sensor evaluation was to ensure that the best source was being used for each controller. Consequently, a pressure transducer was added to the sensor suite to improve altitude data for the altitude controller. A sideslip or beta vane was added for future use by a sideslip controller. The project did not progress far enough to include the sideslip controller design as originally intended. The new sensors were fully calibrated and the results are included in Appendices A and B.

This report documents the heading controller design process from the initial design to the final flight test phase. In order to fully integrate the new heading controller into the FMS, the development had to include extensive evaluation and testing of the airspeed and altitude controllers designed by previous thesis students [Refs. 2 and 3]. This ensured both compatibility in performance and consistency in operating controls. The flight test results in this report include the most significant data collected from

onboard sensors to demonstrate sensor accuracy as well as performance of all three of the completed controllers (airspeed, altitude, and heading).

The design and test equipment include:

1. A highly modified FROG UAV (Fig. 1.1) from the U.S. Army.
2. The $MATRIX_X$ Product Family of software tools developed by Integrated Systems, Inc.
3. A ground station built at the Naval Postgraduate School (NPS) using commercially available computer and communication equipment.

In order to provide the reader with a better understanding of the critical equipment necessary to make this effort possible, a brief description is provided in Chapter II.

Ultimately, the goal of this project is to field a computer-controlled autopilot that can support autonomous flight and future image processing guidance systems. The conclusions and recommendations of this thesis are aimed at making that goal a reality through the continued design evolution of a flight management system (FMS) using the RPS.

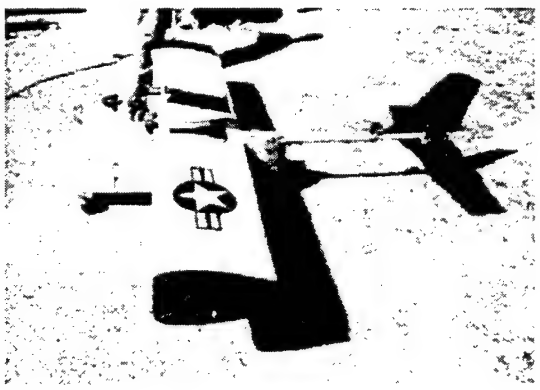


Figure 1.1: FROG UAV

II. RAPID PROTOTYPING SYSTEM

The purpose of a RPS is to aid the systems engineering process by providing a set of integrated tools that allow the engineer to quickly design, test, and implement a control concept. The RPS developed by the Naval Postgraduate School's Department of Aeronautics and Astronautics utilizes the MATRIX$_X$ Product Family of software tools developed by Integrated Systems, Inc. (ISI) of Sunnyvale, CA. Figure 2.1 illustrates how the different MATRIX$_X$ tools are integrated and the functionality each provides. Komlosy, Froncillo, Hallberg, Zanino, and Allen [Refs. 2-6] provide additional information about the RPS and its application to UAV control design.

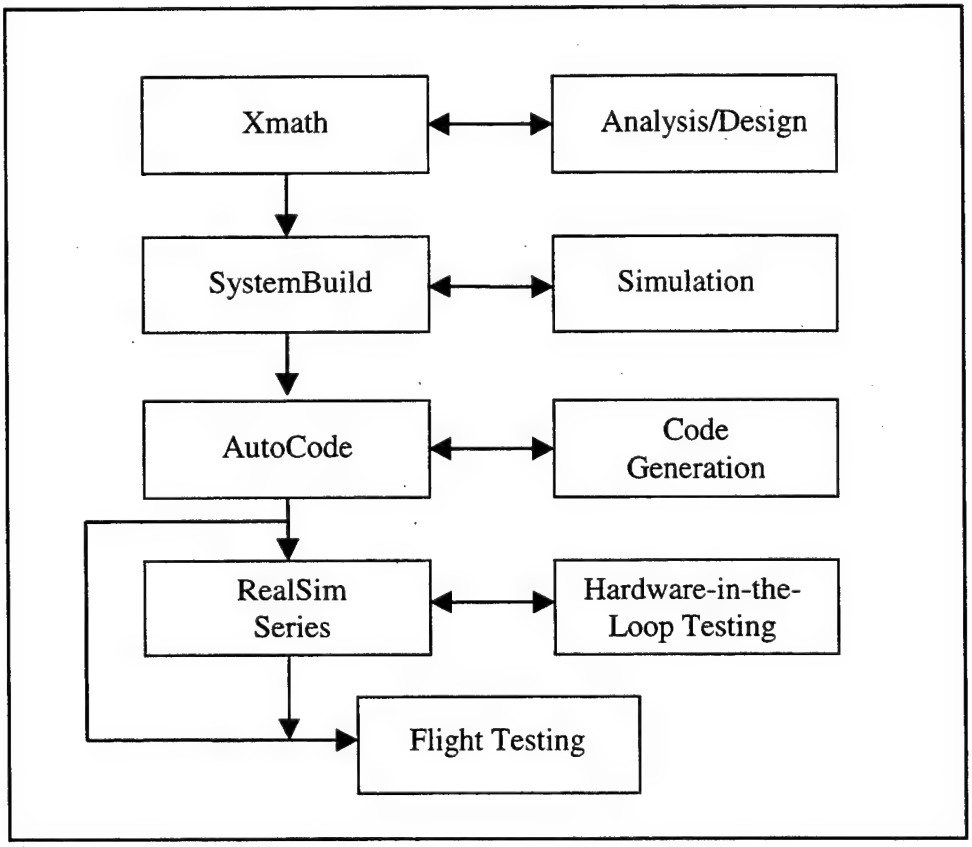


Figure 2.1: MATRIX$_X$ Product Family [Ref. 7]

A. SOFTWARE TOOLS

The MATRIX$_X$ Product Family provides an integrated set of software tools for the development of control systems. The functions of each component of the rapid prototyping software set are summarized below. ISI provides a detailed set of manuals for all the software tools [Ref. 7] and Froncillo provides an excellent tutorial in his Master's Thesis [Ref. 3].

1. RealSim GUI

The RealSim Graphical User Interface (GUI) shown in Fig. 2.2 provides overall control of the MATRIX$_X$ rapid prototyping tools and steps the user through the design process. The GUI also controls other utility functions such as data acquisition and data reduction.

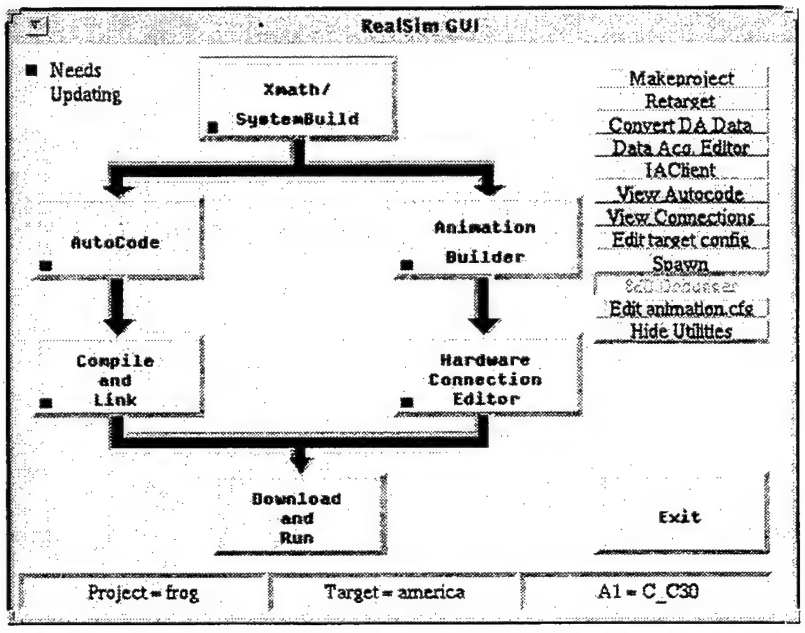


Figure 2.2: RealSim GUI

2. Xmath/SystemBuild

Xmath/SystemBuild is a software tool similar to MATLAB/Simulink. Xmath is the computational engine providing many built-in analysis and control simulation functions. SystemBuild is the graphical, interactive program that uses both built-in and user-defined blocks as modeling system elements. SystemBuild also provides extensive simulation functions.

3. AutoCode

The most powerful and time saving tool of the $MATRIX_X$ products is AutoCode. Shown on the left-hand path of the RealSim GUI (Fig. 2.1), AutoCode uses the real time code file created by SystemBuild to generate high level C-code.

4. Compile and Link

Once the code has been generated, it can be sent to a host computer via file transfer protocol (FTP) by selecting the Compile and Link button. The host computer compiler generates the object code and the link produces the executable code for the processor.

5. Interactive Animation Editor

The right-hand side of the GUI steps the engineer through the design and connection of the input/output (I/O) interfaces for the control system. The Interactive Animation (IA) Editor enables the user to design and build a graphical interface with the control system to allow real-time inputs as well as display of selected outputs during ground simulation and in-flight testing.

6. Hardware Connection Editor

The Hardware Connection Editor (HCE) is used to associate the system I/O's with specific types of external I/O hardware. Many different external I/O devices are available from ISI and are provided complete with compatible RealSim drivers. The HCE is

configured to recognize the available I/O boards and allows functions associated with those boards to be selected.

7. Download and Run

The final step is to "Download and Run" by selecting the bottom button on the GUI (Fig. 2.2). This will load the executable code into the target processor and prepare it for real-time operation. The IA Client Control Window and the upper level user IA interface will appear on the workstation screen (Fig. 2.3). The IA Client Control Window enables the computer operator to start and stop the real-time controller. Once "Start Controller" is selected, the IA interface windows are active and allow commands to be input and data output displays to be observed. The Client Control Window also allows the system variables selected with the Data Acquisition Editor to be recorded.

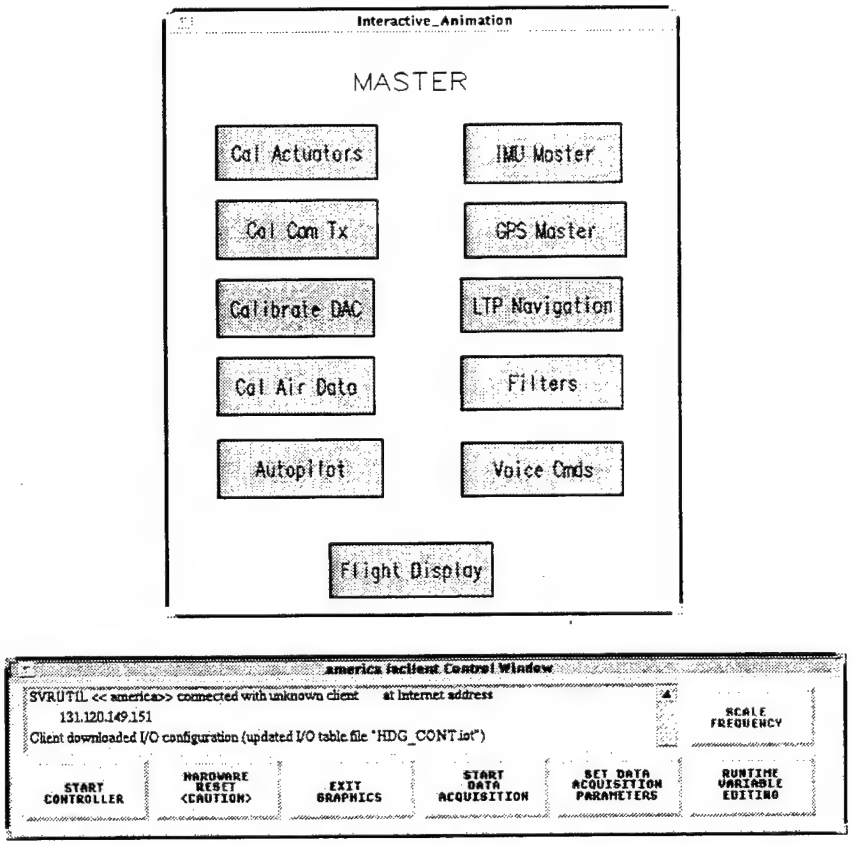


Figure 2.3: Real-Time Control Windows

B. HARDWARE

The hardware portion of the RPS is designed around readily available commercial equipment and can be broken down into two major categories: the Ground Station and the FROG. Komlosy [Ref. 3] provides a detailed discussion of the hardware. The essential components are summarized below.

1. The Ground Station

The "portable" Ground Station consists of four major components (Fig. 2.4):

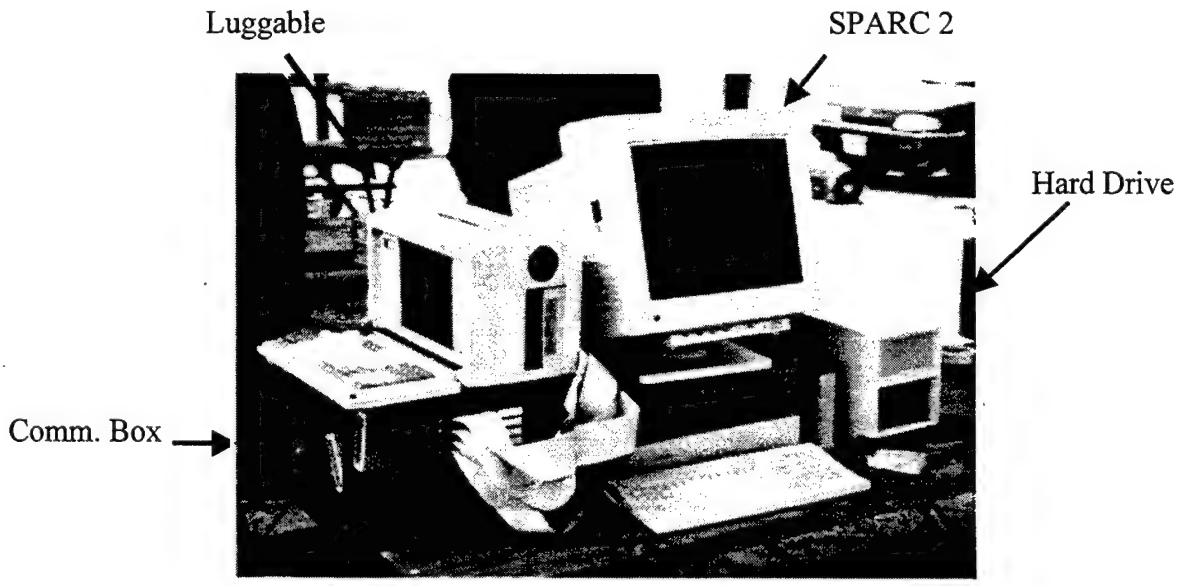


Figure 2.4: RPS Ground Station

a. The SPARC 2 Workstation

The SPARC 2 workstation executes all the $MATRIX_X$ software tools described in Section A of this chapter. This computer is utilized for everything from initial development to actual control of the aircraft during flight test.

b. Luggable PC/IP Modules

The Luggable PC unit (Fig. 2.5) contains the host processor (AC-100) and real-time hardware controller (AC-100 Model C30). The host computer handles FTP, compile, link, and download functions of the system described in Section A of this chapter. The C30 board holds four I/O boards called "IP" modules. Once "Download and Run" is selected on the RealSim GUI, the C30 executes the controller code and provides commands to the IP modules and the IA screens. Komlosy [Ref. 3] describes the I/O configuration of the four IP modules in detail.

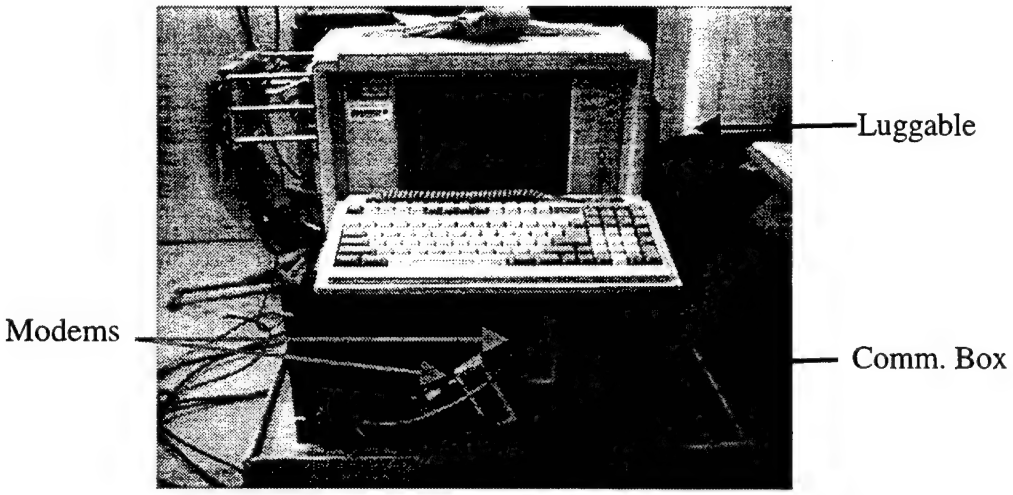


Figure 2.5: AC-100/Communication Box

c. Communication Box/Antennas

The Communication Box (Fig. 2.5) contains all the equipment necessary to transmit and receive data and control signals between the Ground Station and the UAV. This includes two spread spectrum radio frequency (RF) modems [Ref. 8], a Global Positioning System (GPS) and a Futaba pulse width modulation (PWM) receiver identical to the one in the aircraft. Digital data from the Inertial Measuring Unit (IMU) in the FROG are received by one of the modems. The second modem receives aircraft GPS data and transmits GPS differential corrections to the FROG in a full duplex mode. The

extra Futaba receiver permits monitoring and recording the actual command signals being transmitted to the FROG from the Master Futaba controller (discussed later in Subsection 2). Allen [Ref. 6] discusses in depth the use of Differential GPS (DGPS) with the FROG UAV. The antenna array shown in Fig. 2.6 has two helix antennas, one for each modem, and a "puck" antenna for the DGPS. The Communication Box is connected to the array by three coaxial cables, one for each antenna.

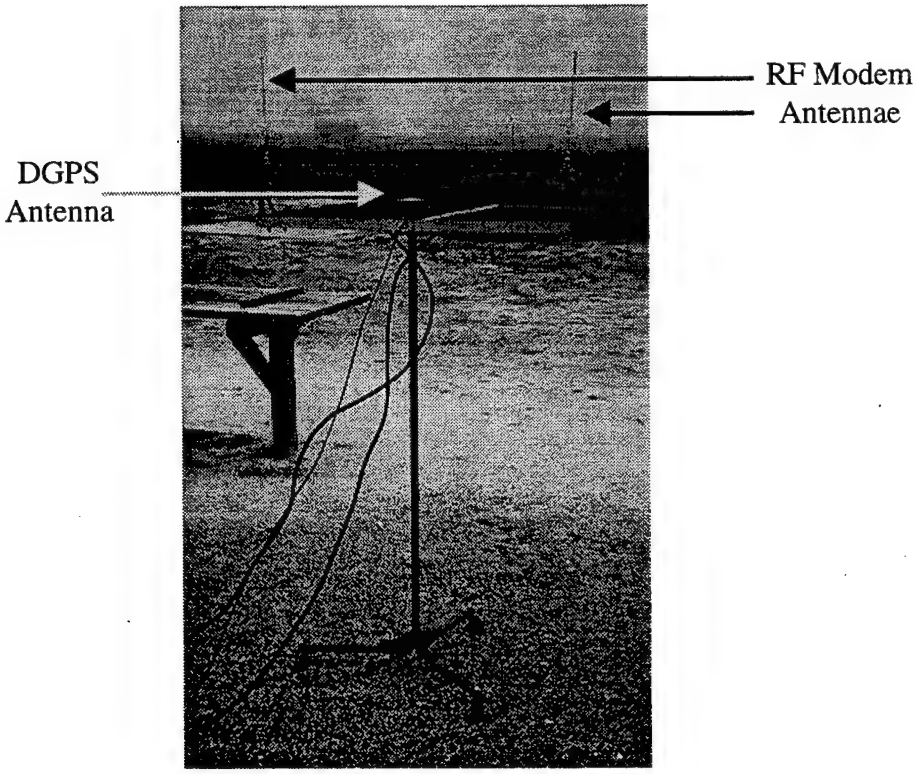


Figure 2.6: Antenna Array

d. Futaba RC Controllers

The aircraft is controlled using a standard Futaba radio control (RC) (Fig. 2.7), which is used by RC model airplane pilots. An identical RC controller has been rewired to take inputs from the C30 and allow computer control of the UAV using standard PWM signals. This transmitter is then connected by a trainer cable to the pilot's Futaba controller and operated as a "slave" in the "trainer" mode. This mode, developed

to train novice RC pilots, allows the slave transmitter to command the UAV as long as the pilot holds the trainer switch engaged.

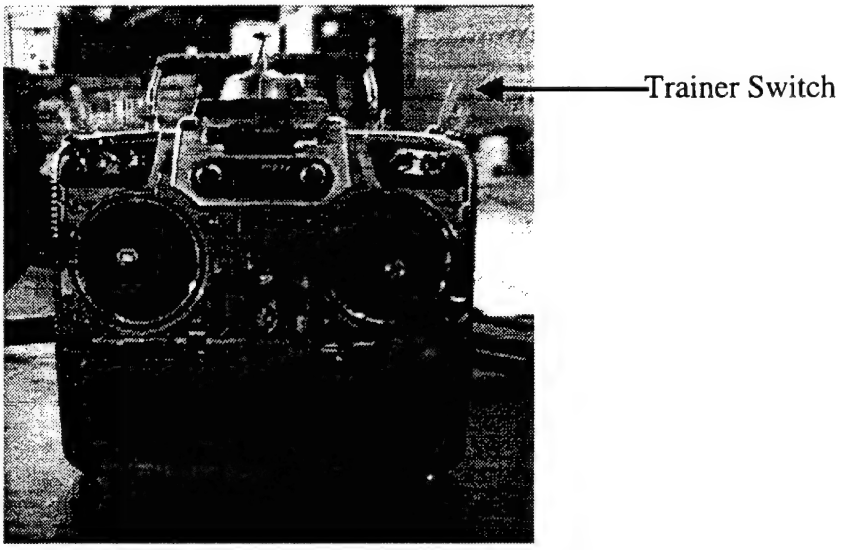


Figure 2.7: Futaba Controller

2. FROG UAV

The FROG flight vehicle (Fig. 2.8) was obtained from the U.S. Army's TEXCOM Experimentation Center at Fort Hunter Liggett, CA. The UAV is a high wing monoplane with the engine mounted on a pylon atop its twelve-foot wing span. Originally wire-guided, the UAV has been converted to the same radio control system commonly used by RC model aircraft enthusiasts [Ref. 2]. The control system also includes an autopilot with its own yaw and climb rate sensors, which allow the pilot to fly by essentially commanding turn and climb rates rather than control surface movements. This is the easiest way to fly the FROG except during takeoffs and landings, when the reduced control authority in the autopilot mode is insufficient.

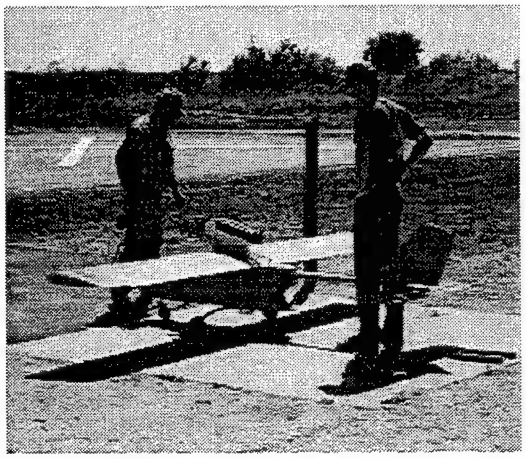


Figure 2.8: FROG UAV

a. Sensors

The onboard sensor suite currently includes a full pitot-static system, consisting of separate static and total pressure transducers, which output analog voltage signals to the IMU 16 bit analog-to-digital (A/D) converter. The sideslip, or beta, vane is attached to a single-turn potentiometer mounted on the pitot boom. Its analog voltage signal also goes to the IMU.

b. Watson Inertial Measuring Unit (IMU-600D)

The onboard IMU is a solid state gyro system, which performs functions similar to an attitude gyro and a slaved heading gyro. The angular rate sensor signals are coordinate transformed and then integrated to produce attitude and heading outputs that reflect normal aircraft attitude coordinates. The attitude and heading signal errors are calculated by comparing the attitude and heading with two vertical reference pendulums and a triaxial fluxgate magnetometer. These errors are filtered and are used to adjust sensor biases so that the long-term convergence of the system is to the vertical references and the magnetic heading. Compensations for centrifugal forces and velocity changes are used to improve overall stability and accuracy.

The IMU allows the user to input up to four analog data inputs and a velocity input that can then be added to the RS-232 serial data output. This allows the

system to act as a data acquisition unit for other vehicle information. For this project, the velocity, altitude and sideslip sensors were connected to the IMU. [Ref. 9]

c. Motorola Encore Global Positioning System

The GPS operates in a differential mode utilizing corrections from an identical GPS located in the ground station. The GPS receive antenna is located on the tail boom just forward of the vertical and horizontal tail surfaces.

d. DGR-115 FreeWave Modems

Two spread spectrum RF modems made by FreeWave Technologies, Inc. are located in the FROG center payload bay. They transmit digital data from the IMU and GPS to the ground station. The GPS modem also receives differential corrections from the Ground Station GPS in a full duplex mode. Both links operate at 9600 Baud. Reference 8 contains additional information on the two wireless transceivers.

III. FLIGHT MANAGEMENT SYSTEM (FMS)

The manual control of a UAV in flight requires precise audio/visual cues to the pilot from the vehicle's engine and airframe. Depending on the vehicle's size, beyond about one half nautical mile from the pilot on the ground, the critical flight parameters such as attitude, airspeed, altitude, and rate of changes, can no longer be observed. Therefore, the UAV's mission range is restricted to a relatively short line-of-sight distance between the pilot and the aircraft. In order to extend the useful range of a UAV, an FMS, which allows the pilot to monitor and control the basic flight parameters independent of visual range, is necessary.

The control of three basic flight parameters is essential in all aircraft maneuvering and navigation tasks: airspeed, altitude and heading. In order to reduce oscillations in aircraft with a lightly damped Dutch Roll mode, sideslip (β) or yaw angle may also need to be controlled. Thus, the FMS being designed for the FROG will ultimately include controllers for all four of these flight parameters. The airspeed and altitude controllers had already been designed by Komlosy [Ref. 2] and Froncillo [Ref. 3] respectively (former postgraduate students at NPS). For completeness, a summary of their controller designs is provided below. The design of the heading controller is discussed in Chapter IV. Due to time constraints, the project did not progress far enough to include a sideslip controller design. However, a sideslip vane was installed, calibrated and tested to support the future design work (see Appendix A).

A. AIRSPEED CONTROLLER

The airspeed of the FROG is controlled via throttle movement of its engine. The throttle is actuated by a Futaba servo. Unlike the elevator and ailerons, the throttle can not be controlled through the autopilot. Therefore, direct control of the throttle is necessary. The major software components of the airspeed controller are shown in Fig. 3.1 in block diagram format. The design requirements were:

1. Seamless Transition - no large fluctuations when turned on or off.
2. Zero Steady State Error - tracking errors tend towards zero in light winds.
3. Bandwidth - wide enough for tight tracking, but narrow enough to prevent engine stalls.
4. Stability Margins - 6 decibels (dB) Gain and 45° Phase margins.

The controller has two modes of operation: open loop (OL) or closed loop (CL). For both modes, the computer operator enters a speed change desired in knots.

In the OL mode (Fig. 3.2), the airspeed controller commands a fixed throttle position by adding the speed change command converted to equivalent throttle position change to a reference throttle setting signal. The actual Futaba command signal being sent to the FROG throttle is referenced at the time the trainer switch is activated. No actual airspeed referencing or tracking is performed. An adjustable gain is provided at the input to permit optimizing the gain margin during flight tests.

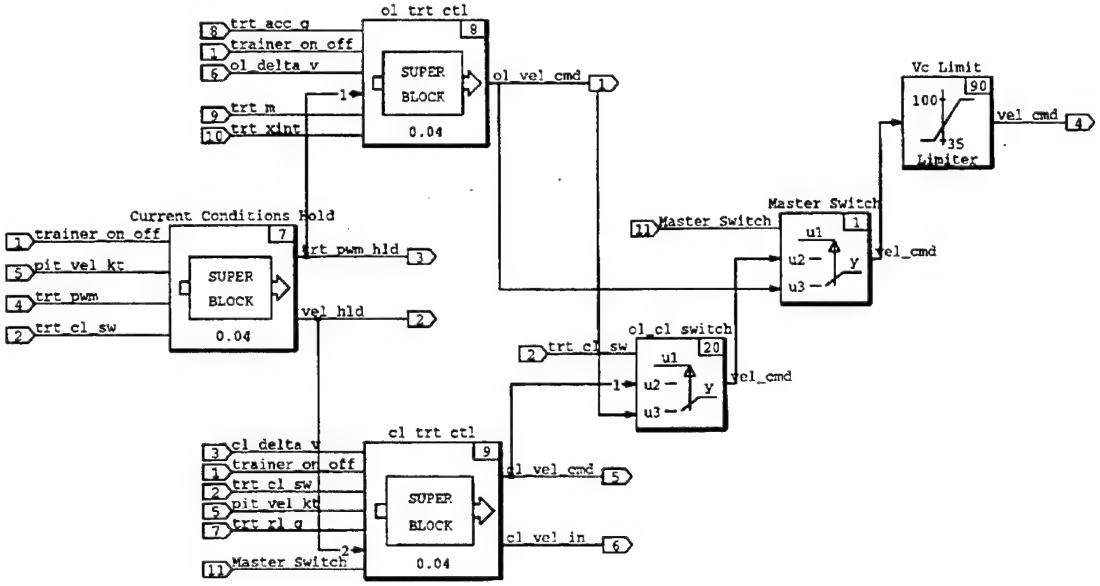


Figure 3.1: Airspeed Controller

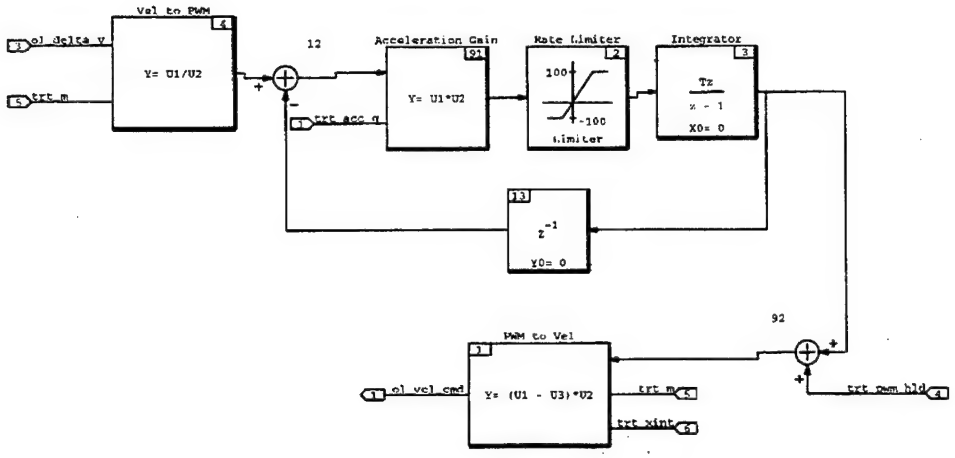


Figure 3.2: OL Airspeed Controller

In the CL mode (Fig. 3.3), the actual airspeed is referenced at the time the trainer switch is turned on and added to the speed change entered. The airspeed controller compares this calculated airspeed with actual pitot-static airspeed feedback to produce a commanded velocity. A Proportional-plus-Integral (PI) controller design was used to achieve the required specifications. The commanded velocity is converted to an equivalent analog voltage before being sent to the slave Futaba controller, where it is converted to a PWM signal and transmitted to the FROG.

There are three switches connected to the airspeed controller: the Trainer switch, the CL switch and the Master switch. Only the Trainer switch is required for OL control. However, for CL control all three switches need to be on. The CL switch allows for individual selection/deselection of the FMS controllers, and the Master switch permits simultaneous selection/deselection of all FMS controllers.

A "wind-down" loop is included at the output of the CL integrator to force the output back to its initial value when the CL switch or Master switch is off. This prevents turning the CL mode on while the output is still at a previous state. An adjustable gain is provided at the input to permit optimizing the gain margin during flight tests. An

adjustable gain is provided between the PI output and the wind-down loop to permit optimizing the gain margin during flight tests.

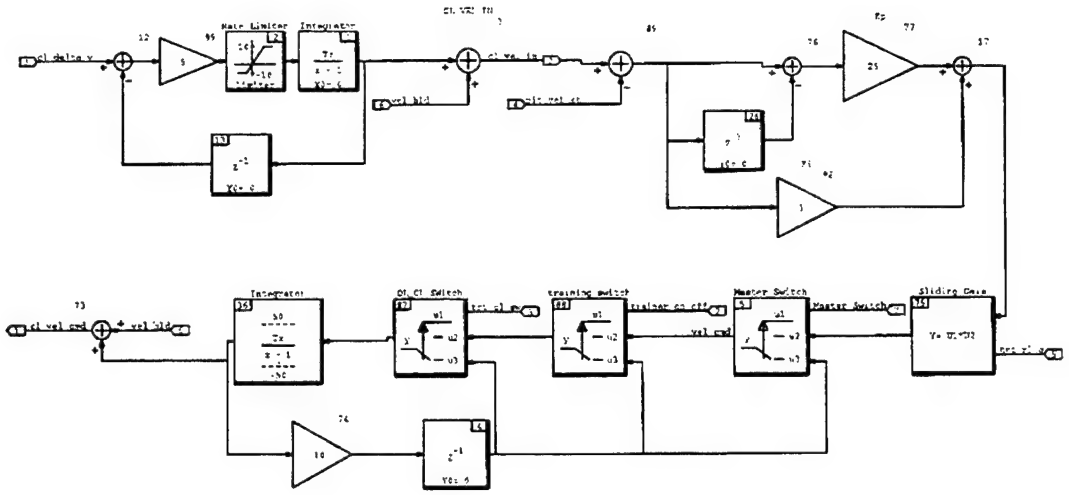


Figure 3.3: CL Airspeed Controller

The following limits are designed into the airspeed controller to ensure no unsafe airspeeds or throttle movements are commanded:

1. OL/CL command input limited to ± 50 kts change.
2. OL throttle command rate limited to ± 100 µsec/sec signal change.
3. CL speed change command rate limited to ± 10 kts/sec.
4. CL integrator output limited to ± 50 kts.
5. Velocity command output (OL & CL) limited between 35 and 100 kts.
6. PWM command signal limited from 1300 to 2100 µsec (equivalent to ~ 40 to 70 kts).

B. ALTITUDE CONTROLLER

The altitude can be controlled either directly through the elevators or indirectly through the autopilot. Control using climb rate commands through the autopilot was chosen as the easiest and safest method to implement, due to the stability already offered by the autopilot. The design requirements were:

1. Seamless Transition - no large fluctuations when turned on or off.
2. Feedback system stable.
3. Zero Steady State Error - tracking errors tend towards zero in light winds.
4. Maximum overshoot (M_P) to a step command of 20%.
5. Rise time (t_r) of 30 sec for a step command of 100 ft.
6. Stability Margins - 6 dB Gain and 45° Phase margins.

The controller has two modes of operation: OL or CL. For the OL mode, the computer operator enters a climb rate desired in feet per minute (fpm), which is converted to an analog voltage output to the Futaba slave RC controller. There it is converted to an equivalent PWM signal and transmitted to the FROG. No actual altitude or climb rate referencing or tracking is performed.

For the CL mode (Fig. 3.4), the computer operator enters a desired altitude change in feet. The actual pressure altitude is referenced at the time the trainer switch is turned on and added to the altitude change entered. The altitude controller compares this calculated altitude with actual pitot-static altitude feedback to produce a commanded climb or descent rate in fpm. A Proportional-plus-Integral-plus-Derivative (PID) controller with "delta implementation" design was used to achieve the required specifications. As in the OL mode, commanded climb rate is converted to an equivalent analog voltage before being sent to the slave Futaba controller, where it is converted to a PWM signal and transmitted to the FROG.

As in the case of the airspeed controller, three switches control functioning of the altitude controller. The Trainer switch and the Master switch are the same switches for all controllers and perform the same function. Each controller has its own CL switch to allow for individual selection/deselection of the altitude controller.

The following limits are designed into the altitude controller to ensure no unsafe altitudes or climb rates are commanded:

1. OL command input limited to ± 2000 fpm.
2. CL command input limited to ± 500 ft.

3. PWM command signal limited between 1000 to 1800 μsec (equivalent to approximately -1450 fpm to 2550 fpm).

A pressure transducer was installed to provide accurate altitude feedback for CL tracking. Appendix B contains the details on the transducer installation and calibration.

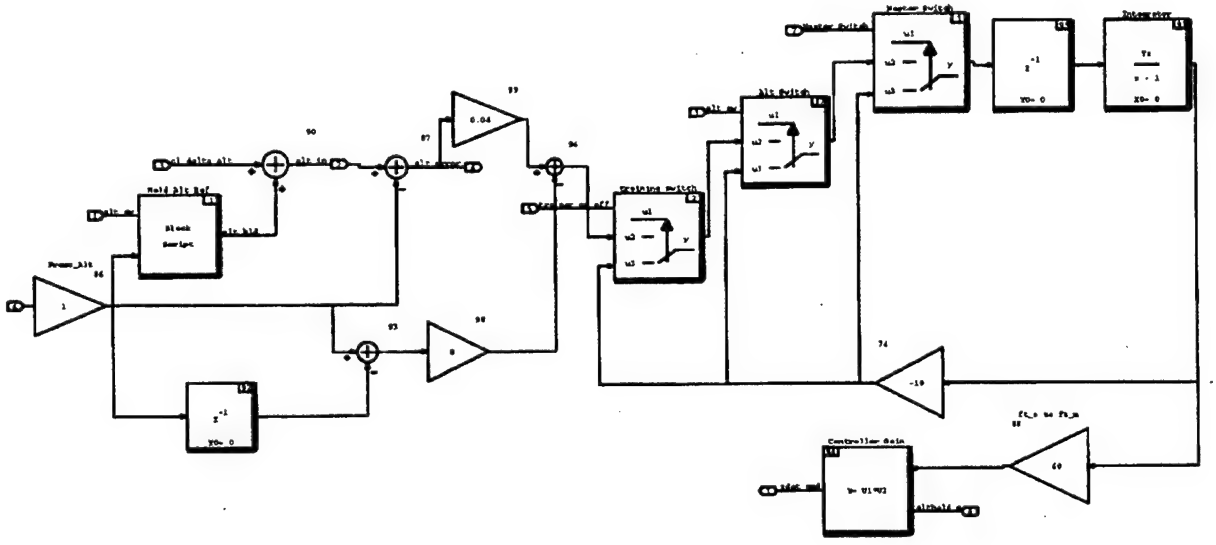


Figure 3.4: CL Altitude Controller

IV. HEADING CONTROLLER

A. DESIGN REQUIREMENTS

An OL turn rate control had already been implemented into the FMS. The operator enters a desired turn rate in degrees per second (deg/sec), which is converted to an analog voltage and sent to the slave Futaba controller, where it is converted to a PWM signal and transmitted to the FROG. No heading or yaw rate feedback is provided, and the accuracy of the command depends entirely on how accurate the conversions formulas are (see Appendix C).

To add a heading controller to this UAV, several methods were available, all of which use a heading error input transformed by the controller to a desired turn command. The turn can be controlled by either using aileron commands as the desired control output, a combination of aileron and rudder commands, or turn/yaw rate commands to the autopilot. As done for the altitude controller, the last method was chosen as the one with lowest risk and easiest to implement. The design requirements were:

1. Seamless Transition - no large fluctuations when turned on or off.
2. Feedback system stable.
3. Zero Steady State Error - tracking errors tend towards zero in light winds.
4. Maximum overshoot (M_P) to a step command of 20%.
5. Rise time (t_r) of 20 sec for a step command of 45° heading change.
6. Stability Margins in control and command loops - 6 dB Gain and 45° Phase margins.

B. MODELING THE AIRCRAFT

The first step in the design process is to create a Plant model of the aircraft, autopilot and control surface actuators as shown in Fig. 4.1 in block diagram form. The aircraft/autopilot model was developed in SystemBuild by Papageorgiou [Ref. 10]. The

second order actuator model was developed as part of a class project for AA 4342, Advanced Controls for Aerospace Vehicles.

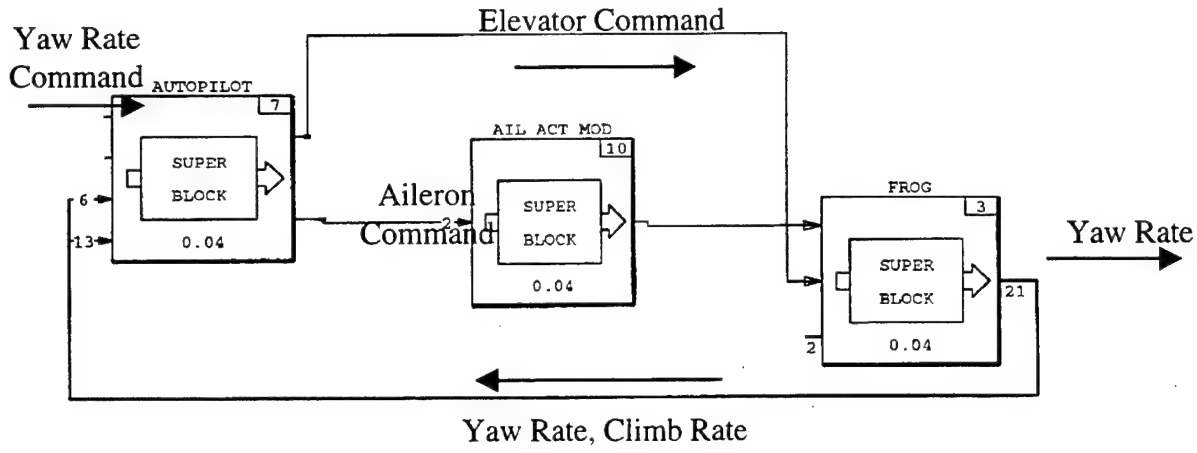


Figure 4.1: FROG, Autopilot and Actuator Model (Open Loop Plant)

To determine the bandwidth available for yaw rate control, the nonlinear model, consisting of the FROG, actuator and autopilot dynamics, was trimmed and linearized about a typical flight condition. Since the FROG has to be flown within ½ mile of the pilot to maintain good visual cues, frequent turns are required during flight tests. Therefore, for controller analysis and synthesis, the nonlinear model was trimmed and linearized about the flight condition characterized by 52 kts true air speed, 5 deg/sec yaw rate, zero flight path angle (γ) and zero sideslip (β).

Tables 4.1 and 4.2 show the eigenvalues, with their associated damping ratios, and frequencies of the FROG model and the FROG/Autopilot model respectively. Note that the FROG has two unstable modes (at 0.0025 radians per second (rad/sec) and 0.15 rad/sec) that are stabilized by the autopilot. The first unstable mode is due to the coupling of the yaw and bank angles, since the trim point is in a turn. The second unstable eigenvalue represents the FROG's divergent spiral mode. It can be seen from these tables that the FROG is lightly damped in all oscillatory modes with a Dutch Roll damping ratio of 0.15 and frequency of 3.8 rad/sec.

Eigenvalue	Damping	Freq. (rad/sec)
0	-1.0000	0
0	-1.0000	0
0	-1.0000	0
0.0025	-1.0000	0.0025
0.1510	-1.0000	0.1510
-0.0724 + 0.4118i	0.1731	0.4181
-0.0724 - 0.4118i	0.1731	0.4181
-0.5478 + 3.7365i	0.1451	3.7764
-0.5478 - 3.7365i	0.1451	3.7764
-4.1985	1.0000	4.1985
-2.5277 + 3.3848i	0.5984	4.2245
-2.5277 - 3.3848i	0.5984	4.2245

Table 4.1: FROG Model Eigenvalues

Eigenvalue	Damping	Freq. (rad/sec)
0	-1.0000	0
0	-1.0000	0
0	-1.0000	0
0.0001	-1.0000	0.0001
-0.1858	1.0000	0.1858
-0.2360 + 0.6166i	0.3575	0.6602
-0.2360 - 0.6166i	0.3575	0.6602
-2.1899	1.0000	2.1899
-0.7001 + 3.5176i	0.1952	3.5866
-0.7001 - 3.5176i	0.1952	3.5866
-1.4026 + 3.3193i	0.3892	3.6035
-1.4026 - 3.3193i	0.3892	3.6035
-4.2497	1.0000	4.2497

Table 4.2: FROG/Autopilot Model Eigenvalues

The open loop yaw rate frequency response is displayed in Fig. 4.2 via a Bode plot, where it can be determined that the yaw rate control bandwidth equals about 1.2 rad/sec.

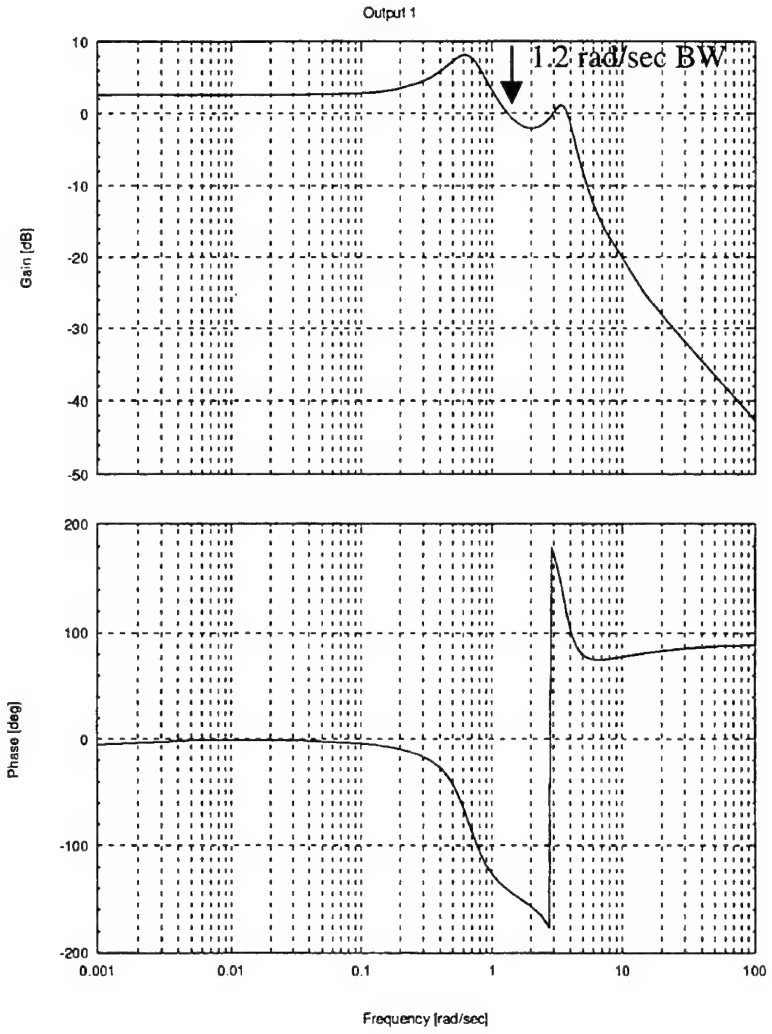


Figure 4.2: Yaw Rate Control Bandwidth

C. DESIGNING THE CONTROLLER

1. Proportional-plus-Integral Controller

The next step is to start the design with a Proportional-plus-Integral (PI) controller, which ensures a steady state error of zero. This is an iterative process employing various methods with the controller gains being the design knobs used to meet the response time, overshoot, and bandwidths requirements. Figure 4.3 shows the basic

PI controller with heading error as the input and yaw rate command as the output. The transfer function of this controller is:

$$C(s) = K_p + K_i / s,$$

where the final values for the proportional gain, K_p, was 0.128 and the integral gain, K_i, was 0.0064. Unfortunately, Plant dynamics produced low frequency oscillations, not associated with the Dutch Roll mode, due to lightly damped complex poles. Figure 4.4 shows the step response in heading, yaw rate and angle-of-bank for a 45° heading change.

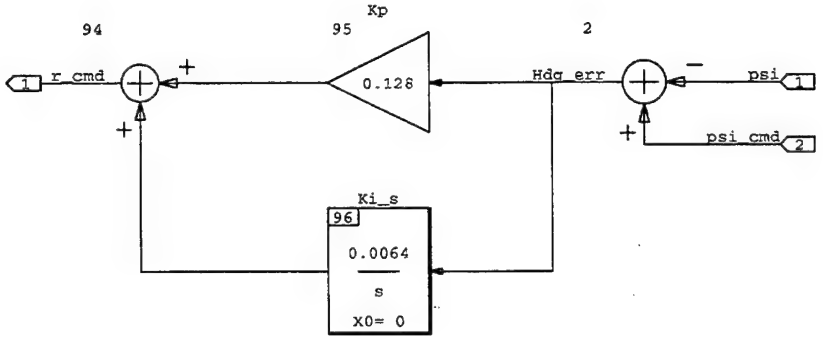


Figure 4.3: PI Feedback Controller Design

2. Proportional-plus-Integral-plus-Derivative Controller

This oscillation drove the design to a Proportional-plus-Integral-plus-Derivative (PID) controller in order to introduce complex zeros to attract the roots and increase the damping of the CL mode. Rather than differentiating heading, however, the yaw rate data from the IMU was used. This avoids amplifying the noise in the heading signal, which is the major disadvantage of derivative control action. The transfer function of the PID controller is:

$$C(s) = K_d \cdot s + K_p + K_i / s$$

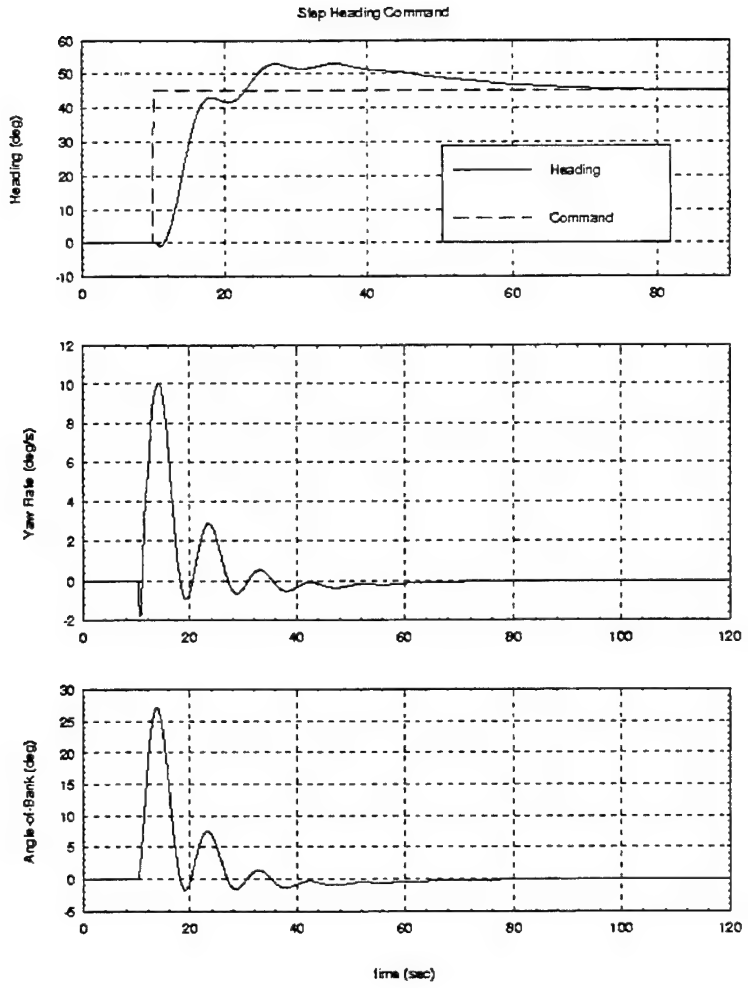


Figure 4.4: Step Response for the PI Controller

The initial constant values tested were for a damping ratio (ζ) of 0.8 and a natural frequency (ω_n) of 0.1 rad/sec (well below the yaw rate control bandwidth):

$$K_p = 2\zeta\omega_n = 0.16$$

$$K_i = \omega_n^2 = 0.01$$

$$K_d = 1$$

The PID design produced excessive overshoots despite all attempts at adjusting the constants. Figure 4.5 shows the step response in heading, yaw rate and angle-of-bank for a 45° heading change. Suspecting the problem may be related to Dutch Roll

excitation, commanded inputs were put through a low-pass filter to avoid exciting the Dutch Roll mode. No improvement in response was noted.

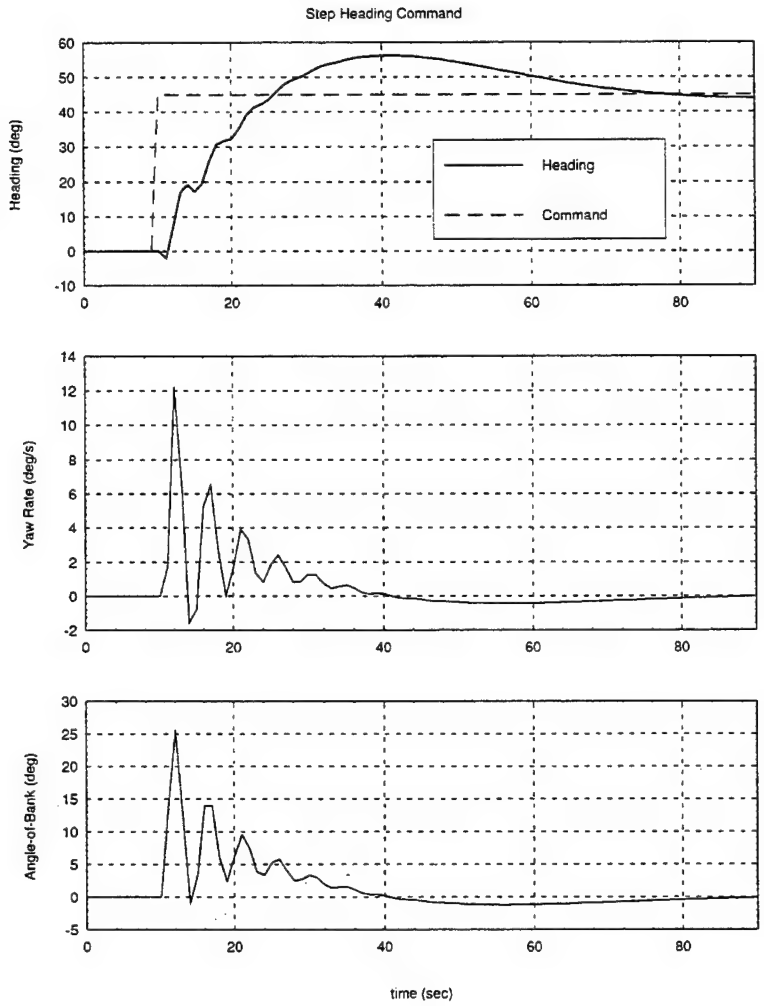


Figure 4.5: Step Response for the PID Controller

3. PID With Delta Implementation

The "Delta Implementation" (Fig. 4.6) was utilized to reduce the excessive overshoot evidenced in the PID controller. This method effectively eliminates the zero added by the derivative controller. This can be shown mathematically by looking at the

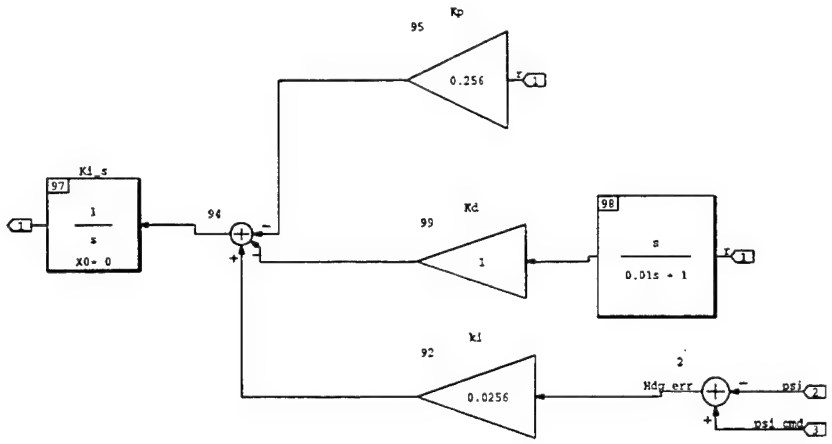


Figure 4.6: PID Controller with Delta Implementation

CL transfer function for the PID controller, where P(s) denotes the FROG, autopilot, and actuator transfer function:

$$\frac{\psi}{\psi_C} = \frac{\dfrac{(K_d \cdot s^2 + K_p \cdot s + K_i)}{s} \cdot P(s)}{1 + \dfrac{(K_d \cdot s^2 + K_p \cdot s + K_i)}{s} \cdot P(s)}$$

Note the zero created by the controller. With the Delta Implementation, the CL transfer function becomes:

$$\frac{\psi}{\psi_C} = \frac{\dfrac{K_i}{s} \cdot P(s)}{1 + \dfrac{(K_d \cdot s^2 + K_p \cdot s + K_i)}{s} \cdot P(s)}$$

Note that the zero has been eliminated, which effectively reduced the overshoot. Also, note that the derivative was approximated by a high-pass filter with a very high cut-off frequency. The rise time, however, was reduced by this design, requiring an increase in the natural frequency from 0.1 to 0.16 rad/sec. This resulted in the final constant values

shown in Fig. 4.6. Figure 4.7 shows the step response in heading, yaw rate and angle-of-bank for a 45° heading change of the final controller configuration. The heading response improvements realized as the controller design evolved from PI to the final Delta implementation are clearly illustrated by Fig. 4.8. The transient response characteristics for the 45° step command are summarized for the three controller types in Table 4.3. It is important to note that in simulating the FROG/autopilot responses to all the different controller designs, a 200 ms time delay was included to duplicate actual signal transport delays between the Ground Station and the FROG.

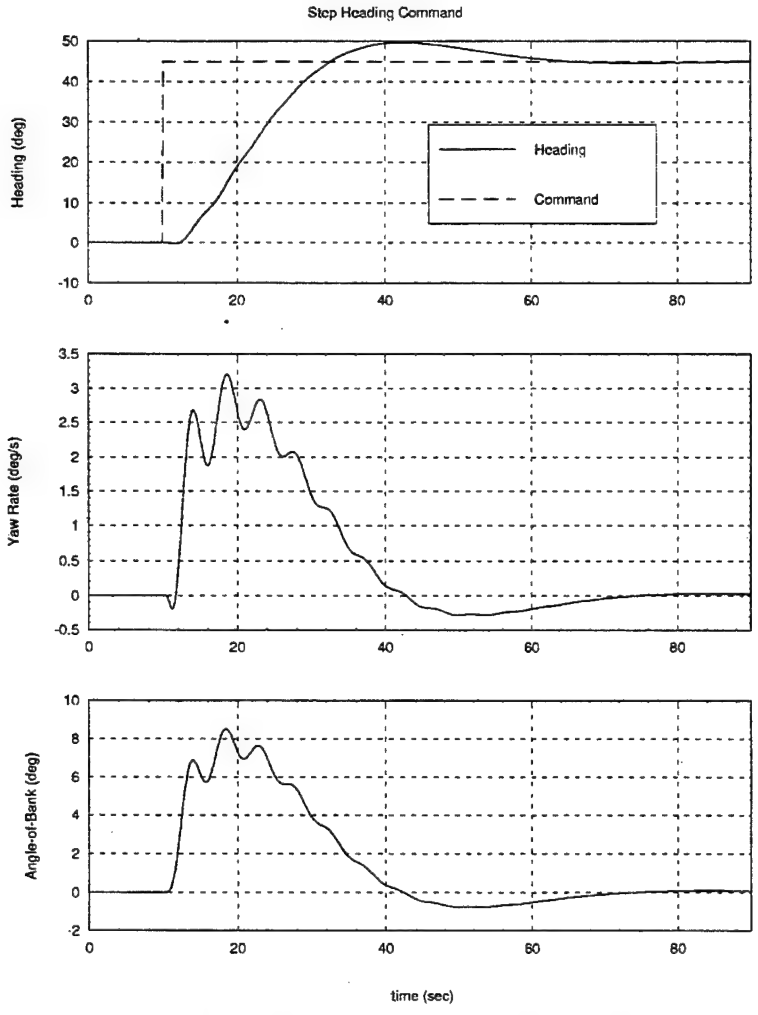


Figure 4.7: Step Response for PID Controller w/ Delta Implementation

In the end, a design compromise had to be made. The requirement for a 20-sec rise time was relaxed slightly to produce significantly less overshoot, which is considered the more important characteristic for heading control of the FROG.

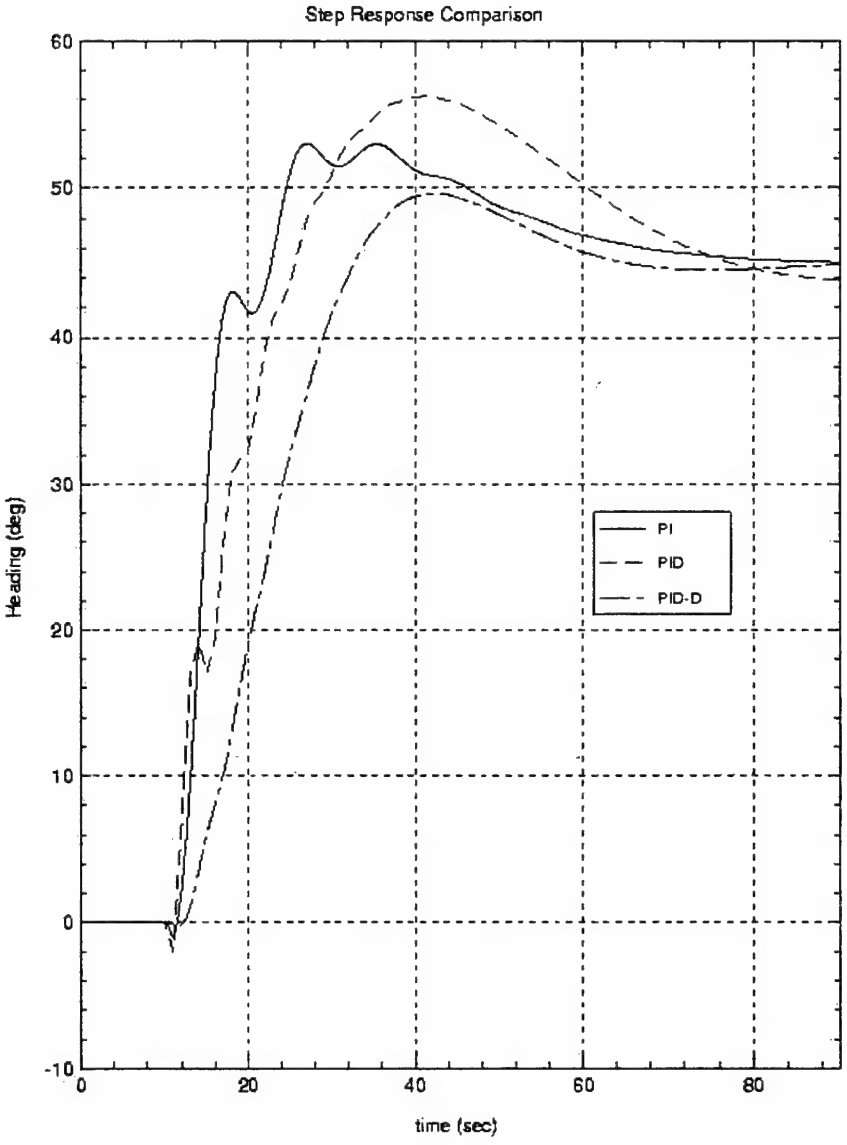


Figure 4.8: Step Response Comparison for Three Controllers

Response Characteristic	PI Controller	PID Controller	PID Controller W/Delta Imp.
Damping Ratio	NA	0.8	0.8
Natural Frequency (rad/sec)	NA	0.1	0.16
Time Constant (sec)	20	12	8
Time Response (sec)	13	26	23
Maximum Overshoot (%)	20	24	11

Table 4.3: Response Comparison for Controllers

4. Control and Command Loop Analysis

To determine the control loop bandwidth and stability margins of the system the root-locus and Bode analyses were done. The loop between the controller and plant was broken as shown in Fig. 4.9 and the system was trimmed and linearized about the same flight condition defined earlier in this Chapter in Section B. The results are shown by the Bode diagrams in Fig. 4.10 to have a gain margin of 27dB and a phase margin of 105°. These are well above the required 6 dB and 45° margins. The control bandwidth of one rad/sec is very close to the same control bandwidth seen on the open loop model.

To determine the command or sensor loop bandwidth the controller loop was closed as shown in Fig.4.11 and the system was again trimmed and linearized about the same flight condition. Table 4.4 shows the eigenvalues with their associated damping ratios and frequencies of the FROG/Autopilot/Controller CL model. Figure 4.12 shows the heading frequency response between the input to the controller and the output from the FROG model. The gain and phase margins are more than acceptable at 25 dB and 50° respectively. The command bandwidth can be see to be approximately 0.1 rad/sec.

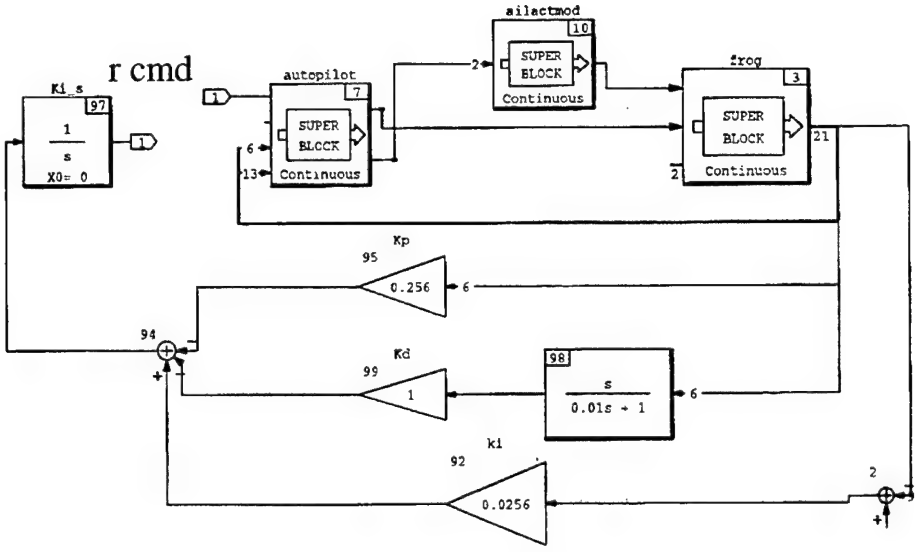


Figure 4.9: Control Loop

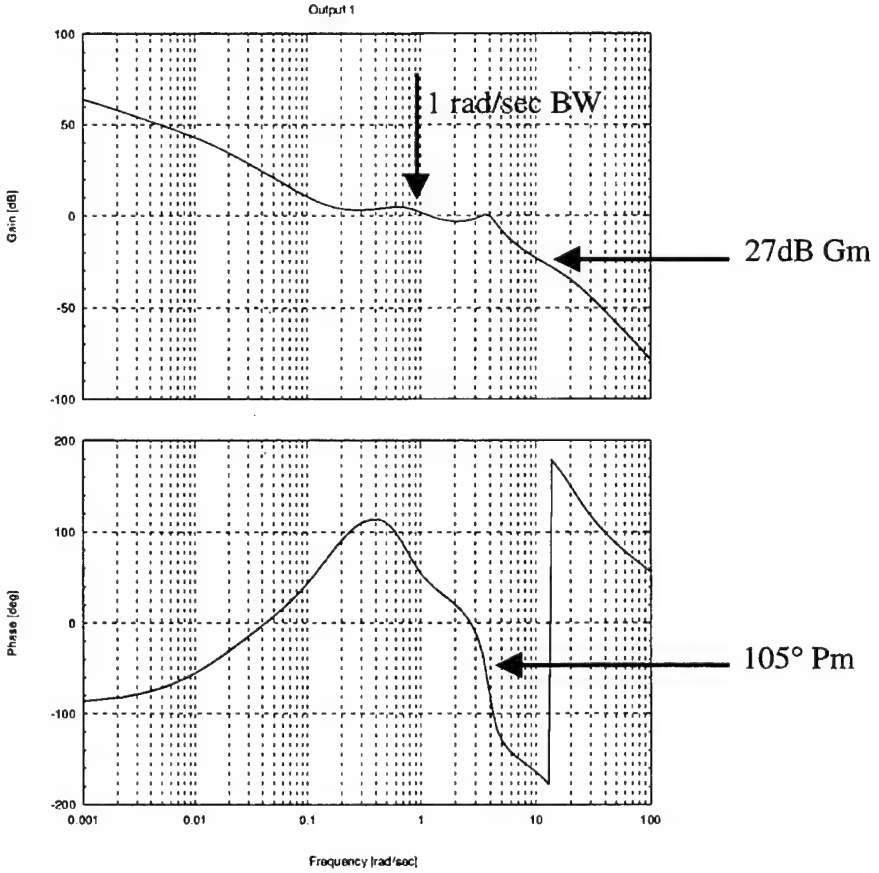


Figure 4.10: Control Loop Bode Plot

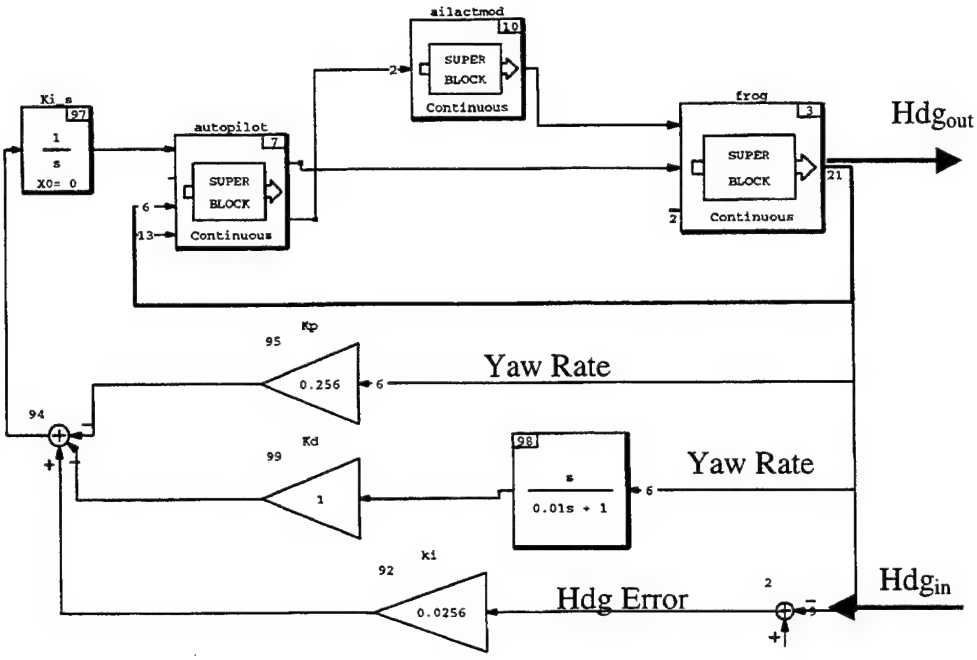


Figure 4.11: Command Loop

Eigenvalue (100×)	Damping	Freq. (rad/sec)
0	-0.0100	0
0	-0.0100	0
0	-0.0100	0
-0.0006 + 0.0010i	0.0054	0.0012
-0.0006 - 0.0010i	0.0054	0.0012
-0.0025	0.0050	0.0128
-0.0064 + 0.0111i	0.0050	0.0128
-0.0064 - 0.0111i	0.0050	0.0128
-0.0195	0.0100	0.0195
-0.0031 + 0.0292i	0.0010	0.0294
-0.0031 - 0.0292i	0.0010	0.0294
-0.0136 + 0.0324i	0.0039	0.0351
-0.0136 - 0.0324i	0.0039	0.0351
-0.0409	0.0100	0.0409
-0.1327 + 0.1523i	0.0066	0.2020
-0.1327 - 0.1523i	0.0066	0.2020
-1.0002	0.0100	1.0002

Table 4.4: Closed Loop Eigenvalues

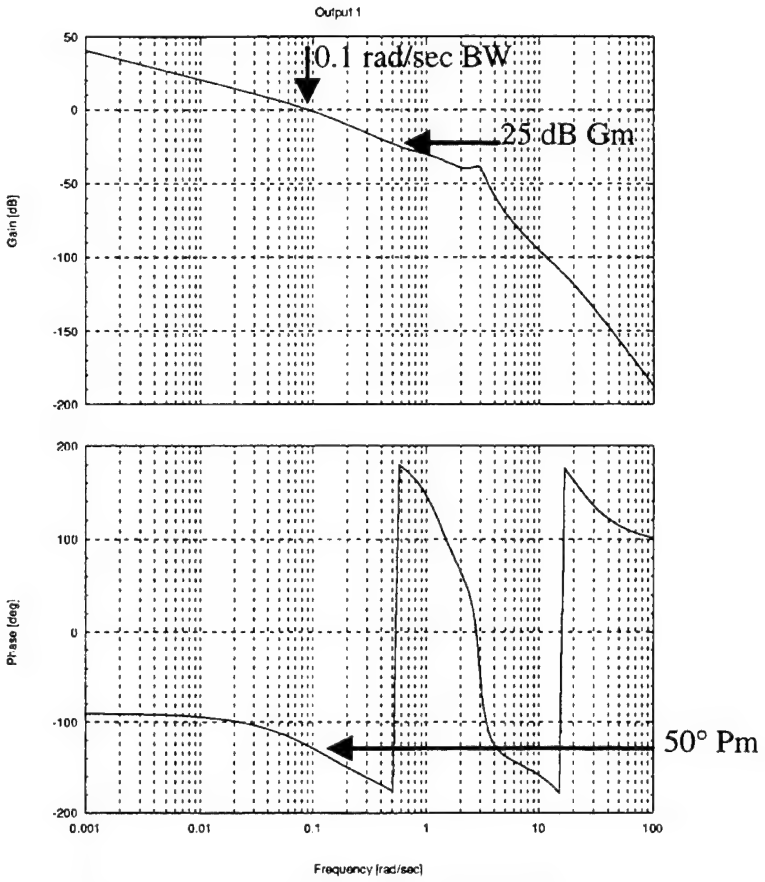


Figure 4.12: Command Loop Bode Plot

D. SIMULATION AND IMPLEMENTATION

After a satisfactory response was achieved, the system model was transformed to discrete-time and tested again to verify satisfactory responses. The final configuration of the controller tested is shown in Fig. 4.13. Step inputs were used to simulate all switch configurations and heading commands, and the dynamic responses were recorded and analyzed. Gaussian noise was introduced to the yaw rate sensor to ensure the controller was not adversely affected by noisy signals. A simple aileron-rudder interconnect (ARI) was simulated by sending a portion of the turn rate command directly to the rudder in the FROG model. This tested the effectiveness of adding rudder commands for turn coordination. The ARI was not included in the flight tests, however, due to time

constraints. Simulation results determined not only that the controller met specifications but also that it operated as expected.

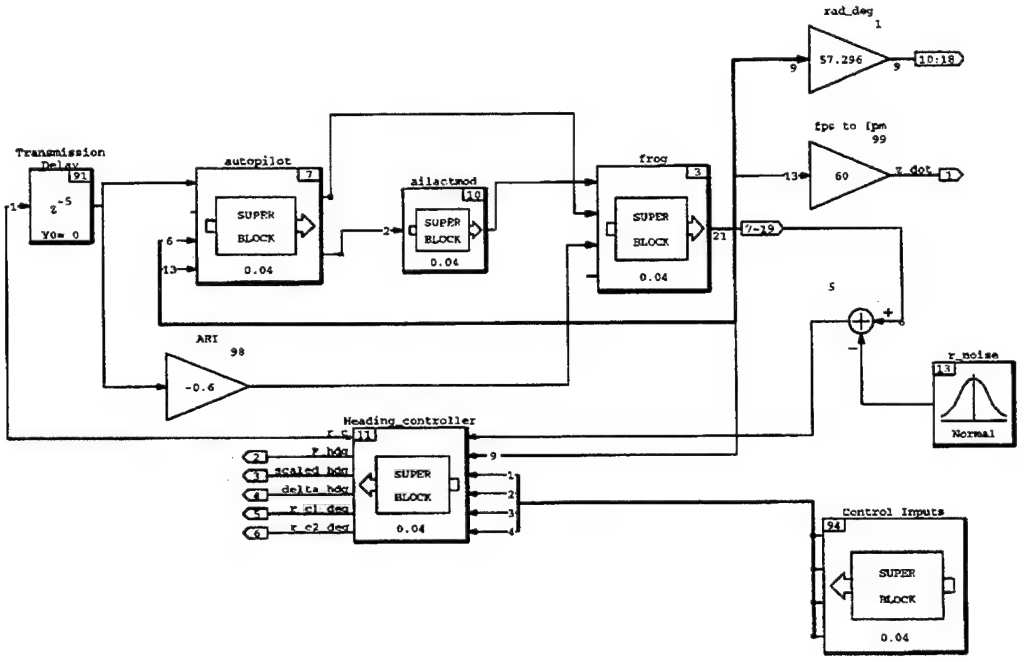


Figure 4.13: Final Model for Heading Control Simulation Tests

It was important to package the heading controller in a "SuperBlock", as shown in Fig. 4.13, with the same inputs and outputs as used in the flight test. This ensured that when the controller SuperBlock was "cut and paste" into the Ground Station FMS software, it could be connected without altering the flight test configuration; thus, the chance of introducing unknowns to the flight test was software minimized. The following is a list of additional modifications required to fully integrate the controller into the FMS.

1. Units Correction

The FROG dynamic software model was built to use radians and rad/sec, but the actual FROG IMU outputs data in degrees and deg/sec. Therefore, any radians-to-degrees conversion gains used in the simulation model were removed.

2. Heading Scaled Correctly

A unique problem to heading controllers is that heading scales differ from source to source: 0° to 360° for a compass, ± 180° for the FROG IMU, or ±∞ for the FROG software model. Therefore, a BlockScript code had to be written (see Appendix D) to automatically scale the input to match the desired 0° to 360° used for the commands and displays of the Ground Station.

3. Switches Installed

As with the airspeed and altitude controllers, three switches were required to turn the heading controller on. See Chapter III for a detailed description of these functions.

4. Wind-Down Loop

As with the other two controllers, a wind-down loop was added at the output integrator that forces the output to its initial value if any of the three switches are off. This prevents initializing the closed loop controller at some previous state.

5. I/O Limits

The following limits were designed into the heading controller to ensure no excessive turn rates are commanded that would result in unsafe angle-of-banks:

1. OL command input limited to ± 20 deg/sec.
2. CL integrator output limited to ± 20 deg/sec.
3. PWM command signal limited from 1150 to 2000 μsec (equivalent to approximately -33 to +22 deg/sec).

6. Interactive Animation Display

The final step in the implementation process is to modify the IA picture to support both the operation of the controllers and the monitoring of critical test data. Figure 4.14 shows the final configuration of the Ground Station's autopilot animation page. The graphical interface provides analog displays such as the altimeter, airspeed and heading gauges in the middle of the screen, as well as digital displays of controller inputs and

outputs. On/Off switches are shown in the lower right corner. Each controller has slider switches that allow both mouse and keyboard entry of commands.

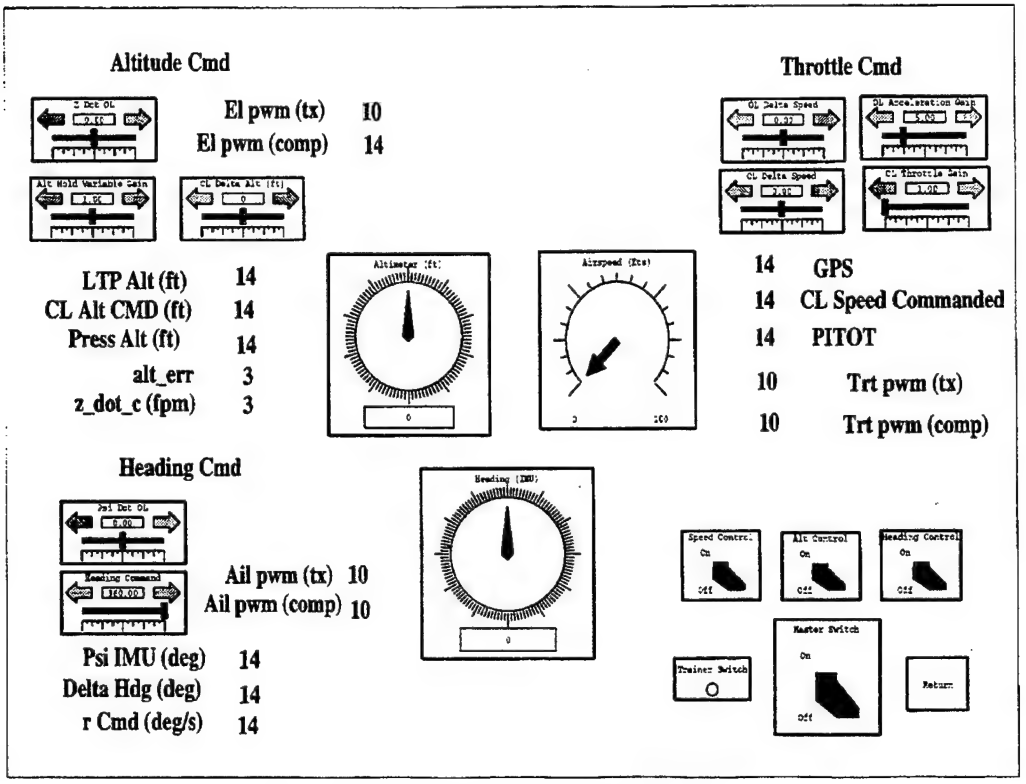


Figure 4.14: Autopilot Animation Page

7. Heading Hold Mode

Since 0° to 360° are used for heading commands, a method had to be developed to permit the operator to command zero change (not 0° heading). It was decided since the IMU output used ±180°, the command 360° would result in a zero turn rate output from the CL controller. Appendix D provides details on the software code used.

E. HARDWARE-IN-THE-LOOP TESTING

Prior to all flight tests, ground testing was conducted in the UAV Lab at the Navy Golf Course. There all systems were powered up, and the latest software was compiled,

linked, downloaded and run on the Luggable PC to verify data transmission/receipt, display operation, and that both software controllers and aircraft flight controls functioned properly. Once it was verified that all three FMS controllers were receiving valid feedback data from the pitot-static system or IMU, the Futaba PWM control signals were calibrated with the voltage outputs from each of the FMS controllers. This was followed by a trim-check of the Futaba RC control boxes by observing aileron, elevator and throttle deflections when the Trainer switch was activated on the Master Futaba control box. If other than slight movements were noted, the Futaba controller trim was adjusted until near zero deflections were evident when the Trainer switch was turned on.

Operational checks of the individual FMS controllers would then be performed in both the open and closed loop modes. The three switches were activated in different orders to ensure no commands were sent unless the appropriate switches were on. The FROG control surfaces were observed to ensure movement in the proper direction. Since the aircraft was static, actual tracking errors and response characteristics for the controllers could not be evaluated on the ground.

V. FLIGHT TESTING FMS

The flight testing was conducted exclusively at an airstrip in Chualar, CA maintained and operated by the Salinas RC Modelers Club. The airfield features a 300 ft asphalt strip positioned in a Northwest/Southeast orientation. The entire Ground Station, FROG and support equipment were packed-up and transported using two U.S. Navy mini-vans. An equipment checklist is provided in Ref. 2, Appendix C. The FROG has no fuel level indications; therefore, flight time was limited to 30 minutes from take-off to landing to ensure ample fuel reserve for varying engine throttle settings.

A total of seven test flights were conducted on four separate days. Due to an IMU heading failure, only limited data was used from the first flight on 7 August 1998. All data runs were used to evaluate sensor data. Dedicated calibration runs, with the RC pilot performing specific steady state maneuvers, were used to verify the relationship between the transmitted PWM signal value and the flight parameter being controlled (airspeed, rate of climb or rate of turn). Dedicated runs were also used to collect performance data on all three controllers in both OL and CL modes. Table 5.1 summarizes the fifty-nine runs, during which data was recorded by the Ground Station. The seventeen runs from the flight with an IMU failure are not included. Note that the number of runs in each category (not bracketed) attempts to identify the primary purpose of the run. The numbers in brackets denote runs, during which more than one FMS controller was actively flying the aircraft (i.e., two or three of the controllers were engaged in either the OL or CL mode).

Run Type		Flight 8/14/98	Flight 8-21-98 #1	#2	Flight 8-28-98 #1	#2	#3	Total
Taxi					1			1
Take-Off		1	1		1	1		4
Landing			1	1	1		1	4
In Chocks							1	1
Turn Calibration			2					2
Throttle Calibration				3				3
OL	Airspeed	1	5		{3}	{1}	{3}	6
	Altitude	{1}			{3}	{3}	{1}	0
	Heading		2		4	1		7
CL	Airspeed	3	{1}		3	{6}	{3}	6
	Altitude	1	{1}	1	{1}	{2}	{5}	2
	Heading	1	4	3	1	8	6	23
Total		7	15	8	11	10	8	59

Table 5.1: Flight Test Summary

A. SENSOR EVALUATIONS

Figure 5.1 is representative of sensor data comparisons that were made for every data run recorded. This particular example was taken during a steady state right turn.

1. Airspeed

The top strip chart compares pitot-static airspeed with GPS velocity in knots. As was seen in all runs, there appears to be about a 5 knot difference between the two sources of airspeed. Some difference is expected due to wind, since GPS actually measures ground speed. However, the bias should reverse directions in the case shown in Fig. 5.1, where the UAV is performing three 360° turns. The constant bias independent of heading indicates that the difference is more likely attributed to calibration error in the pitot-static system. GPS errors would be more random. The static pressure transducer's voltage output is software filtered and then converted to feet per second (fps) using a sixth order approximation obtained by Papageorgiou [Ref. 10] during his development of

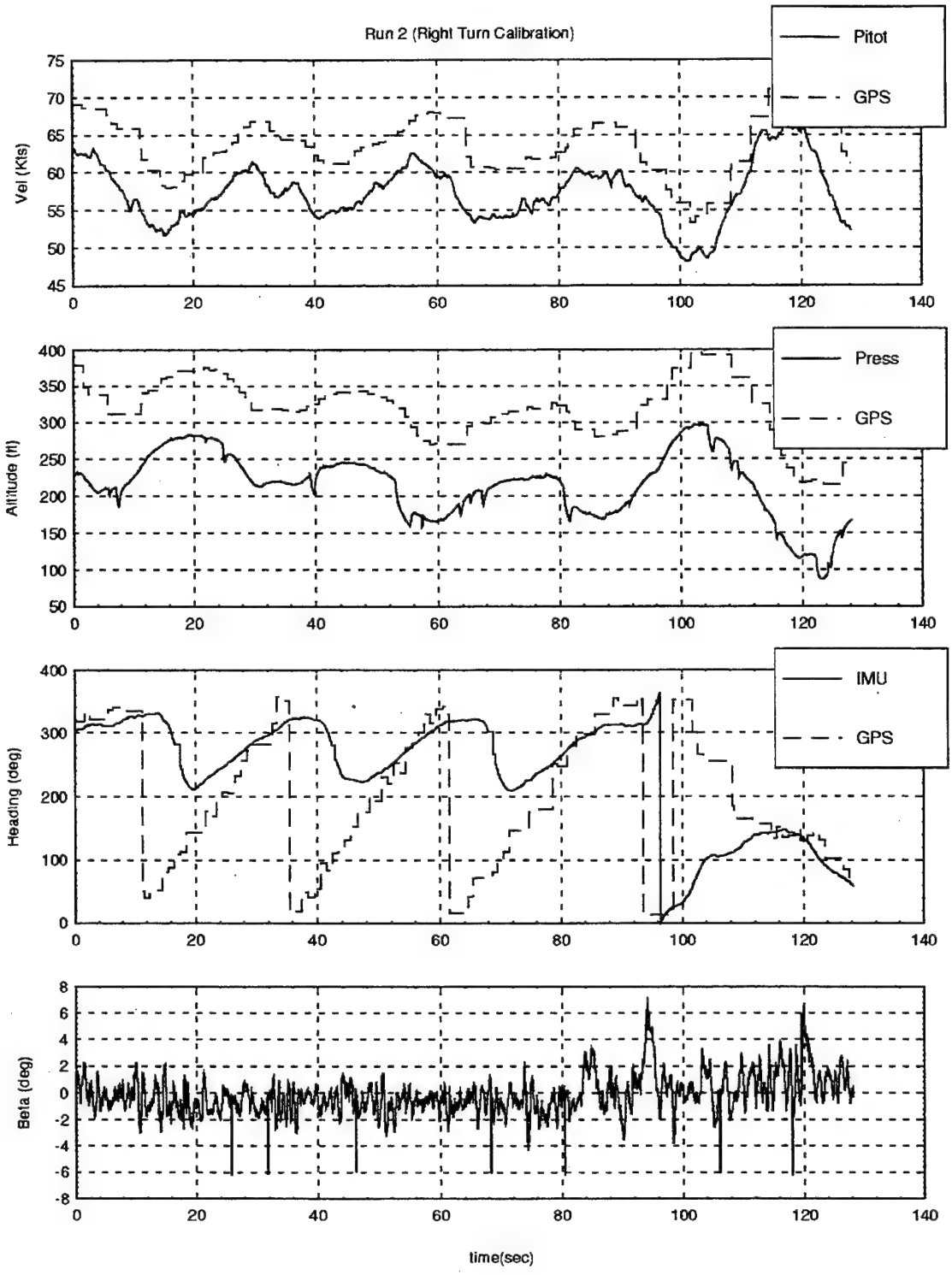


Figure 5.1: Sensor Data for a Right Turn

the dynamic model of the FROG UAV. The low-pass filter was developed by Komlosy [Ref. 2] during his development of the airspeed controller. Time and use may have changed the properties of the pressure transducer and modified the conversion formula slightly. Stationary on the ground, the pitot airspeed generally indicated six to seven knots negative. Calibration checks were not repeated during this project, because the airspeed accuracy was not considered critical to controller performance evaluation.

Either sensor is judged adequate in accuracy for use with this airspeed controller; however, the GPS update rate of one second or more results in a "stair-step" input that can reduce the performance of the CL controller and is difficult to filter. Also, controlling airspeed using GPS ground speed may result in aircraft stalls in high tail wind conditions due to the lower true airspeeds that would be required; therefore, the pitot-static airspeed is the source of choice.

2. Altitude

The second strip chart in Fig. 5.1 compares static pressure and GPS height in feet. Typically, the altitude values from the two different sensors tracked up and down together as seen here with a relatively constant difference; however, the amount of the difference varied greatly from run to run and flight to flight. GPS values were seen that were as much as 300 ft lower and 500 ft higher than the pressure altitude. In all cases, the pressure altitude agreed more closely to visual cues (i.e., when it read zero the aircraft was on the ground and when it read 200 ft the aircraft looked about 200 ft above the ground). The maximum update rate of once per second for the GPS data is again evident in the "stair-step" altitude trace. Some plots showed as much as eight to ten seconds between updates in altitude, although the pressure altitude was indicating a steady change in altitude.

The pressure transducer provides an analog voltage that is hardware filtered in the aircraft and software filtered in the Ground Station. The same software filter used for the airspeed sensor voltage is used prior to conversion to feet. A first order (linear)

conversion is used to calculate the value in feet. The operator inserts an additional constant into the conversion via the interactive interface to compensate for varying barometric pressures. In this way, the altimeter reading can be zeroed on the ground before takeoff to ensure an accurate comparison to GPS height. Appendix B discusses in detail the altitude calibration method. The GPS sensor provides height in meters referenced to the Ground Station GPS. It is then converted to feet unfiltered for display and recording.

It is clear from the data collected that the pressure altimeter is a better source of altitude data than the GPS both for accuracy and for continuous availability. The one to eight second update rates that were observed are insufficient for the altitude controller to perform CL tracking of altitude errors in a dynamic environment. Note the periodic spikes in the pressure altitude trace. If the time scale were expanded, these would appear more like square pulses of random interval. All recorded signals were plotted and compared to these pulses without successful correlation. These pulses are too short (less than ½ second) and too infrequent to have any significant affect on the altitude controller's performance.

3. Heading

The third strip chart in Fig. 5.1 compares the IMU heading with the GPS heading in degrees. The IMU heading data is converted by the Ground Station to a binary format of degrees scaled from -180° to +180°. The GPS heading is provided in degrees scaled from zero to 360°. The heading controller has a BlockScript routine that correctly scales any heading input from zero to 360°.

The GPS heading, while continuously tracking accurately in the correct direction, still exhibits the undesirable "stair-step" data trace with up to 10 seconds between updates. Note that the IMU is unable to track the heading changes to the right near Northerly headings except for the final turn. Instead, the IMU magnetic heading spins in the opposite (left) direction until it intercepts the correct heading and then tracks to the right (increasing heading). This behavior was observed in turns in either direction and

sometimes when going through a Southerly heading. When the IMU headings had been tested on the ground, the IMU accurately tracked all but the extremely fast turns (60-70 deg/sec).

While researching the IMU specifications to determine the cause of the heading reversals, it was discovered that the IMU requires a velocity input to improve its angle-of-bank estimates. Angle-of-bank is in turn used in the IMU filters to better estimate headings while in a turn. This explained why the behavior was not observed in ground tests. When the FROG is simply rotated at zero angle-of-bank, the pendulums were adequate to provide angle-of-bank for filtered heading. The production model of the IMU being used for these flight tests, however, was not wired to accept velocity inputs. A modified IMU could not be available in time for project completion.

The final analysis is that the smooth trace of the IMU headings make it the more desirable to use for heading control. Connecting the airspeed input to a properly wired IMU should correct the heading problem. The GPS heading, while reasonably accurate for steady turns, was unreliable in maneuvering flight, when GPS updates were observed less frequently and heading errors grew unpredictably.

4. Sideslip

The bottom strip chart in Fig. 5.1 shows sideslip angle (β) in degrees. When considering the small magnitude of the oscillations ($\pm 1°$ to $2°$) and the compressed time scale of the strip chart, the signal from the sideslip vane potentiometer appears relatively noise-free and usable. There is no other sensor available on the FROG to compare to sideslip for accuracy estimation; however, this is not considered critical for sideslip control. The zero sideslip reference is the critical target, which the sideslip controller will be designed to track. As long as the zero angle position is known and the potentiometer responds linearly about this point, this sideslip indicator will be adequate.

The light damping in the FROG's yaw was evident in the continuous oscillation of sideslip for all maneuvers. This indicates that a sideslip controller could be extremely useful in coordinating turns and dampening yaw oscillations. The calibration and

conversion of the signals from the sideslip vane potentiometer are discussed in Appendix A.

B. AIRSPEED CONTROLLER

Initial flight tests showed the airspeed controller to be unresponsive and limited in effectiveness in both OL and CL modes. The data collected pointed to three basic problems:

1. The minimum and maximum throttle commands allowed were too restrictive.
2. The throttle trim setting used for calibration was too high.
3. The linear formula used to convert PWM to knots was no longer valid due to a new engine installed.

Consequently, the airspeed controller had insufficient authority to significantly change the UAV's speed within its narrow operating envelope of 35 to 70 kts. The airspeed controller's PWM output limits were changed from 1375-1925 μsec to 1300-2100 μsec. This is still well within the maximum operating limits of the throttle, which are about 1100-2200 μsec. Flight test data were collected to update the conversion formulas (see Appendix C). This data was also used to determine the best trim setting for a middle throttle position. The PWM to volts calibration is now done using a trim value of 1650 μsec, which in flight gives an airspeed of about 55 kts (center of the 40-70 kts airspeed range considered safe). The final three test flights included all of these fixes with the results described below.

1. Open Loop Commands

The first data runs made in a flight usually involved OL controller trim checks where the FROG pilot stabilizes the aircraft in straight and level flight at the center calibrated throttle setting. The Ground Station operator would have the OL command inputs zeroed with the Controller and Master switches off (OL position). When the pilot engaged the Trainer switch the aircraft and autopilot control panel were watched closely for transient responses from the aircraft. If all calibration and conversion constants were

set correctly, little or no change in throttle setting should occur. Since the throttle is directly controlled by referencing the pulse width (PW) of the command signal in the OL mode, the conversion constants between PW value and airspeed have no affect on transients. Experience has shown that the throttle calibration for PW to volts does not change significantly during flight. Therefore, the most likely cause of off-trim conditions existing is that the throttle setting at hand-off may be slightly different from the center calibration position. This can be verified by monitoring the throttle PWM command reading on the Ground Station IA autopilot display.

Figure 5.2 is a good example of a zero speed change command given OL to the FROG during run five, flight one on 28 August 1998. The altitude and heading were steady around 275 ft and 300° respectively. The top strip chart compares the actual airspeed as measured by the pitot-static system with the OL velocity command. The middle chart compares actual throttle PWM signal sent from the Futaba controller with the OL velocity command converted to PW equivalent. In this case, since the operator input is zero, the velocity output of the controller is simply the reference PWM signal value at the time the Trainer switch was activated converted to knots. The bottom strip chart plots the Trainer and Throttle Controller switch positions as "one" for on and "zero" for off. These charts show that the CL throttle controller was off and the Ground Station was controlling the FROG for about 22 seconds OL. The FROG airspeed stayed within three kts of the commanded airspeed and the throttle calibration was within 30 μsec of actual throttle position. This is considered extremely accurate and is equivalent to about one knot of airspeed.

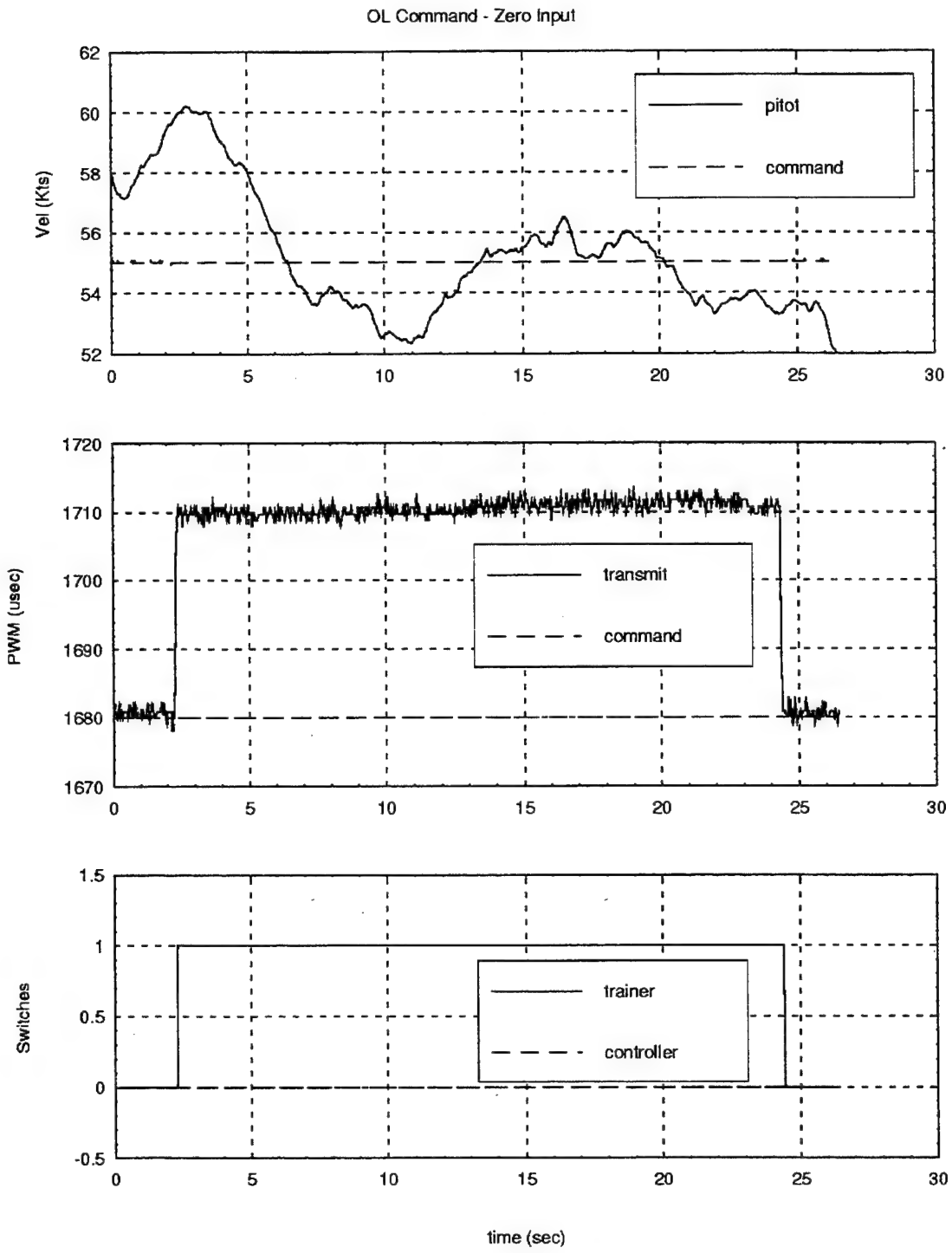


Figure 5.2: OL Airspeed Command

2. Closed Loop Commands

After the OL trim checks were completed satisfactorily, each of the controllers was tested individually and then together. The CL test runs were initiated several different ways to verify proper switch functioning as well as airspeed tracking. Initially the Ground Station operator would keep both Controller and Master switches off with zero commands entered to ensure no controller commands were output prior to the desired time. When the aircraft was in position and the Trainer switch activated, the Controller switch would be activated then the Master switch. Controller outputs were monitored to ensure zero commands occurred until all switches were on. At this point, the operator would input commands to observe the controller's performance. As the flight testing progressed successfully, the operator would preload the desired commands and arm the Controller and Master switches prior to the Trainer switch being activated. This saved time and allowed for more controller tracking time to be recorded before the maneuver had to be aborted for either airspace or fuel considerations.

With the fixes described earlier implemented, the airspeed controller performed acceptably. Figure 5.3 shows an example of good airspeed tracking in the CL mode, while both the altitude and heading controllers were also active in the CL mode. The FROG was in a commanded left turn from 230° to 010° while maintaining 325 ft. A velocity change of +2 kts was entered for this run. The top strip chart compares three different values of velocity in knots:

1. Pitot-static airspeed (solid line).
2. Velocity command output from the airspeed controller (dashed line).
3. Velocity command input to the airspeed controller, which is the sum of the reference velocity and the operator entered velocity change desired (dash-dot line).

For this run the reference velocity held from the time the Trainer switch was activated was 53.8 kts. Hence, the commanded input was constant at 55.8 kts. The output command varied in the proper direction and maintained the actual airspeed within two kts

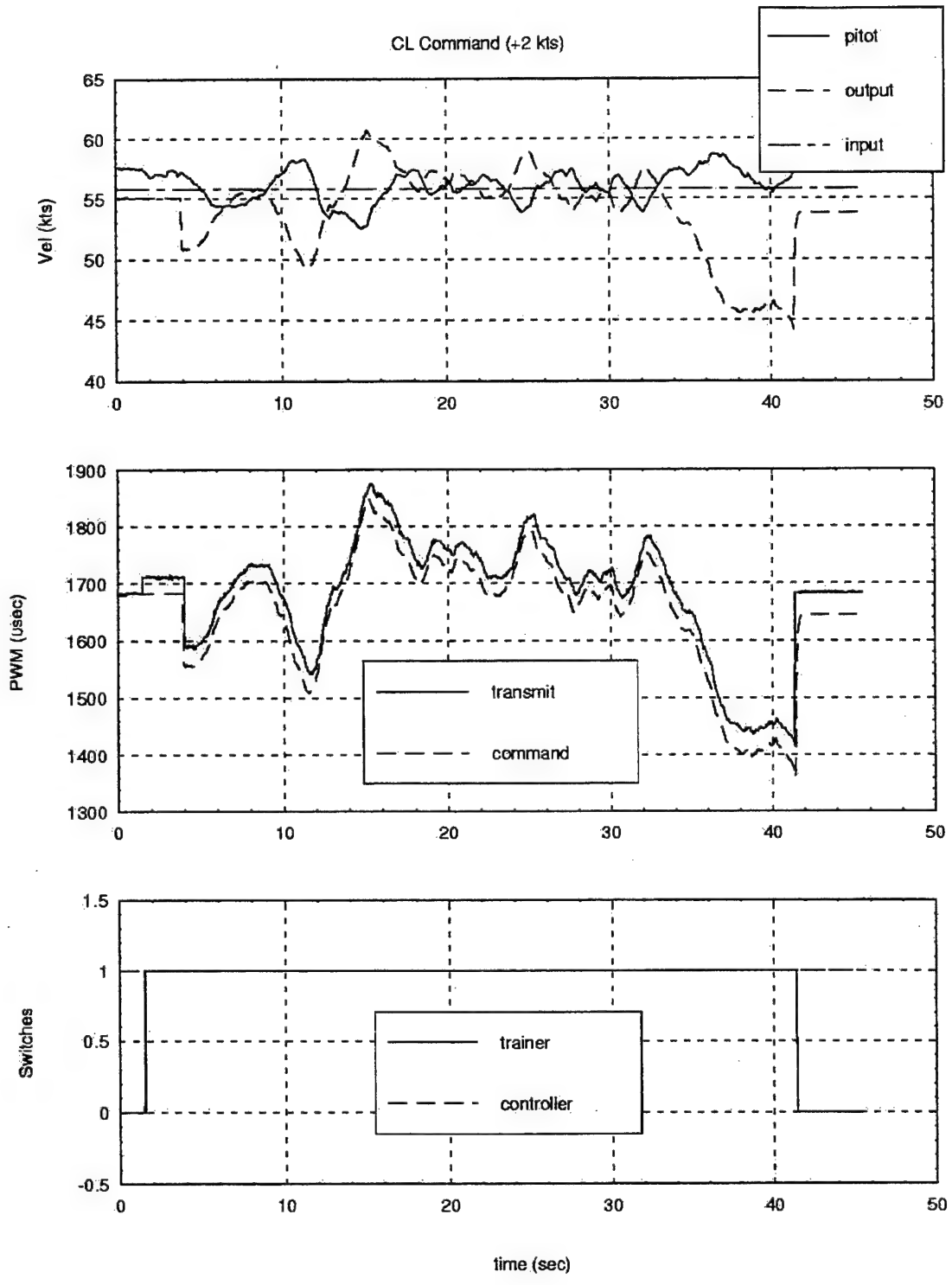


Figure 5.3: CL Airspeed Command

of the desired airspeed. The second strip chart compares the actual throttle command signal from the Futaba controller with the controller velocity command output converted to PW. It shows a 30 μsec calibration error (one-knot equivalent). As expected, this is the same as seen in Fig 5.2 for the OL mode since both runs were on the same flight. As in Fig. 5.2, the third chart plots the Trainer and Throttle Controller switch positions, which were both on for 44 seconds. It is important to note that as a result of IMU heading problems the heading controller was commanding progressively higher angle-of banks (i.e., the FROG was in a wind-up turn). After 32 seconds, the altitude could no longer be maintained by the altitude controller and the aircraft lost over 125 ft before the Trainer switch was released and the pilot took control at 42 seconds. The fact that the throttle position/velocity commanded continued to decrease and maintained actual airspeed within two kts of the commanded 55.8 kts during this extreme maneuver is an impressive demonstration of robustness for the controller.

C. ALTITUDE CONTROLLER

Initial flight tests in the OL mode showed a tendency to command a climb when command inputs were zero. In the CL mode, tracking was in the proper direction, but not always as responsive as expected. The data collected indicated a difference in what the controller output was commanding and what the aircraft was receiving for command signals. This could be attributed to either errors in the formula to convert climb rate commands to PWM equivalent commands or the calibration of PWM to volts. The fpm to PW conversion could have changed because of the extensive rewiring and system modifications the FROG had undergone since tests were last conducted. However, the calibration method appeared sound and could not have been affected. Consequently, additional flight test data were collected to update the conversion formulas (see Appendix C). The new linear approximation formula derived was significantly different in both the slope and the "x-intercept". The change in the "x-intercept" overcorrected the tendency to climb, and resulted in a slight descent with zero command input in the OL mode. The

"x-intercept" value was adjusted during flight until a zero command input resulted in straight and level flight (see Appendix C). However, the increase in the value of the slope effectively increased the gain of the altitude controller (i.e., a specific rate of climb change would result in larger changes in PW of the transmitted signal). This unexpectedly resulted in the CL mode becoming unstable and slowly divergent. This wasn't discovered until the second from the last test flight. In order to complete the final flight safely, it was decided to change the conversion formula back to the original values. The results of the altitude controller flight tests are highlighted below.

1. Open Loop Commands

The primary purpose of the OL altitude control was to maintain a steady altitude. Therefore, most of the OL altitude controller testing was done with zero command input to verify that the conversion and calibration were being done properly about the trim (zero rate) condition.

Because there is no vertical speed indicator (VSI), the pressure altitude data was fed through an integrator with a unity feed back loop. The estimated vertical speed was taken from the integrator input and converted from feet per second (fps) to fpm (see Fig. 5.4). The first recorded altitude value was used as the initial condition for the integrator to avoid an infinite spike at the beginning of the calculations. Note that a gain of one was chosen to give the "VSI" a one rad/sec bandwidth. This was considered sufficiently narrow to filter out any noise in the altitude data without reducing response time significantly.

Figure 5.5 shows the altitude controller data for a "zero command" OL trim check conducted during run one, flight two on 28 August 1998. Engaging the Trainer switch resulted in steady flight parameters of about 55 kts airspeed, 350 ft altitude, and a 275° heading. The top strip chart shows the pressure altitude in feet holding within 25 ft, of 350 ft. The second chart compares the estimated vertical speed with the zero climb rate command. Although the vertical speed trace oscillates up and down considerably due to

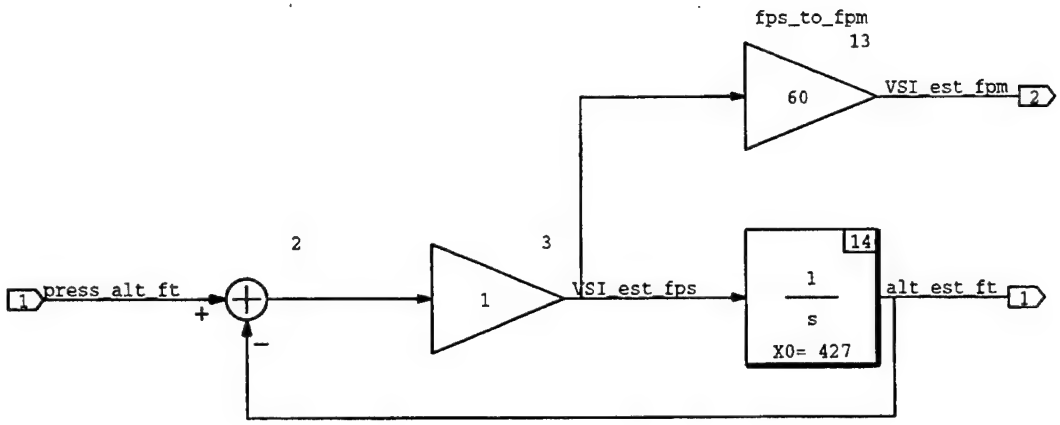


Figure 5.4: Vertical Speed Estimation

unusually noisy altitude data, the mean value is clearly zero. This confirms an accurate conversion formula for fpm to PWM. More specifically this is a result of the "x-intercept" being adjusted to its proper value on the prior flight. The third chart compares the transmitted altitude command signal with the equivalent PW value for zero climb rate. The two traces overlay nicely when the Trainer switch is in the on position as shown in the bottom chart. This confirms an accurate calibration between PWM and volts for the Ground Station. The bottom strip chart shows that the CL controller was indeed off and the OL controller active for 25 seconds.

2. Closed Loop Commands

Due to the airspace restrictions mentioned earlier (both horizontally and vertically), it was impossible to observe the altitude controller's tracking performance for large altitude changes (greater than 100 ft) or over long periods of time (greater than 30 seconds). Consequently, very few data runs were conducted using the CL altitude controller alone. Additionally, the requirement to fine-tuning the conversion formula and the subsequent instabilities induced resulted in very few runs where the altitude controller was considered performing optimally.

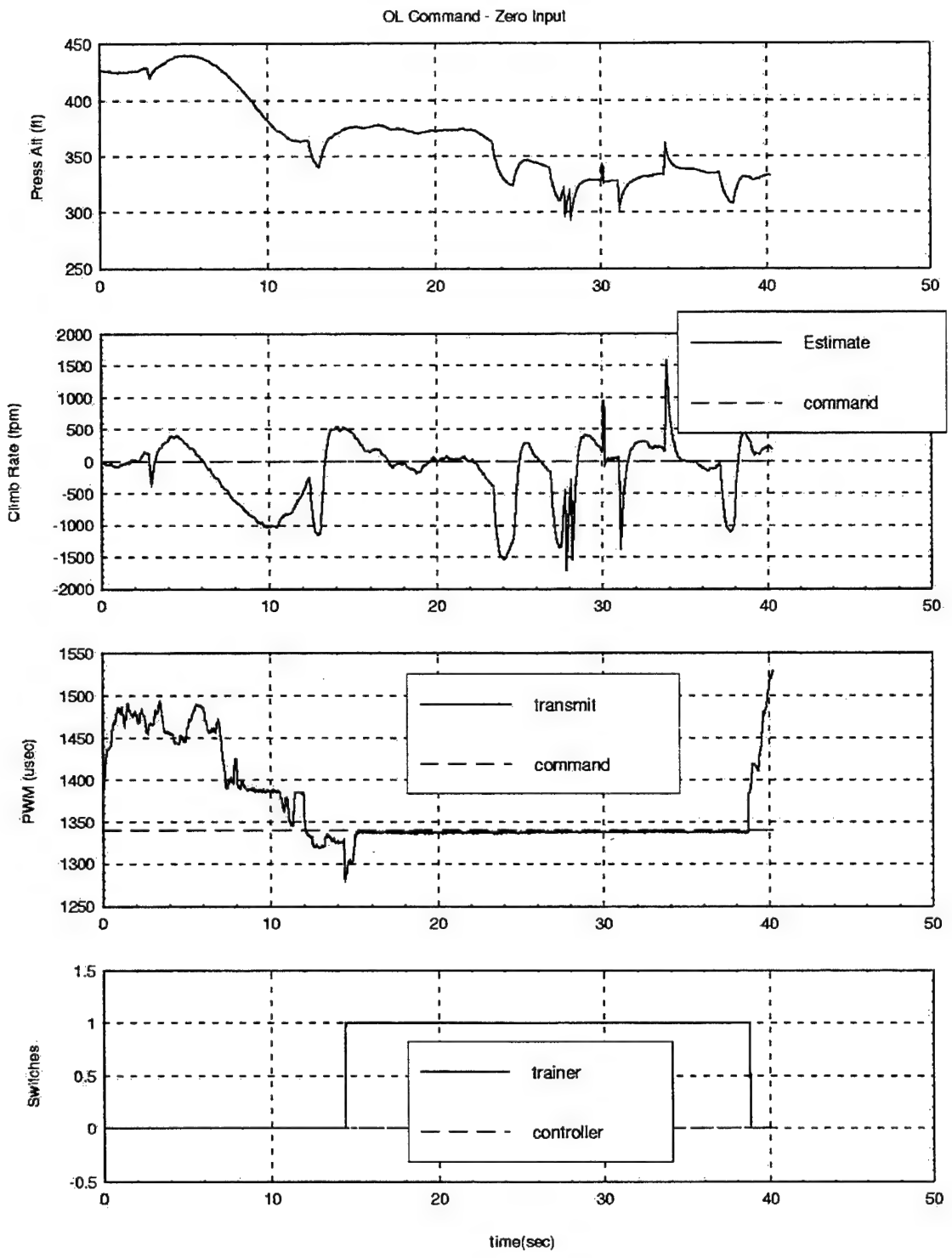


Figure 5.5: OL Altitude Command

On the final day of flight testing, the first two flights uncovered the CL altitude controller's tendency to diverge while tracking altitude in straight and level flight. This was attributed to the effective increase in gain due to the conversion formula update. However, Fig. 5.6 shows extremely accurate altitude tracking in a turn using the same conversions constants. The top strip chart compares the actual pressure altitude with the command input of approximately 327 ft. They are almost identical until near the end of the run where the bank angle and turn rate (greater than 10 deg/sec) exceed the altitude controller or FROG autopilot's capabilities. The second chart shows good comparison between the estimated vertical speed and the commanded climb rate proving that the conversion formula was more accurate for turning flight. This is not surprising since most test data was recorded in a turn. The third strip chart shows the controller's climb rate command converted to PW equivalent overlays the actual signal transmitted to the FROG. Thus, calibration is very close. Finally, the bottom chart confirms that both the Trainer and Controller switches were on for 40 seconds.

During the final flight, when the climb rate command conversion constants were changed back to the original values, the altitude controller did not track as well in turns. It tended to correct more slowly and overshoot the target altitude as shown in Fig. 5.7. However, it no longer exhibited any instability. Therefore, the CL altitude controller is considered adequate to meet its designed intent of acquiring and tracking assigned altitudes as long a the gain is adjusted to the level required for flight maneuvers.

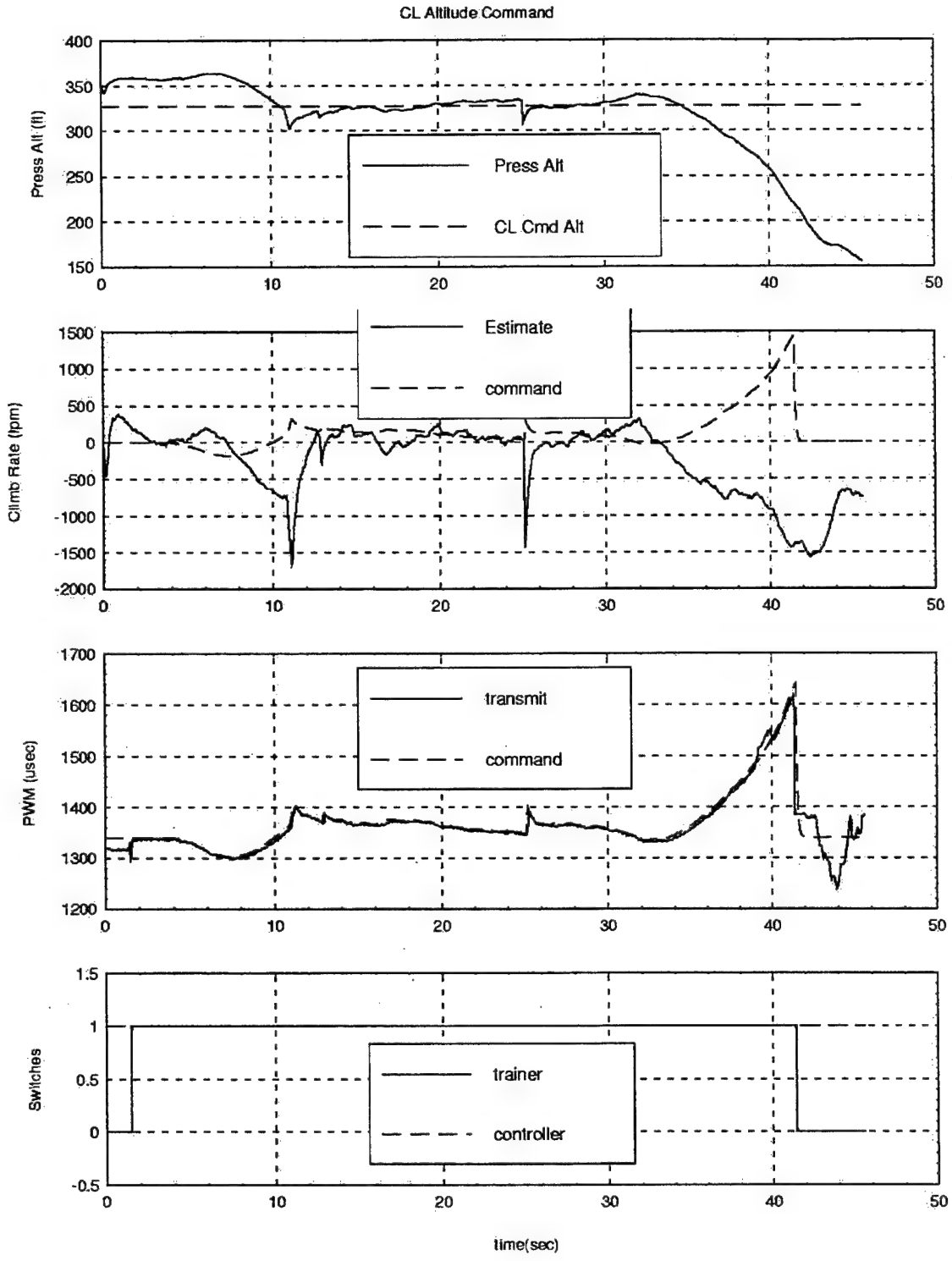


Figure 5.6: CL Altitude Command (New Conversion Constants)

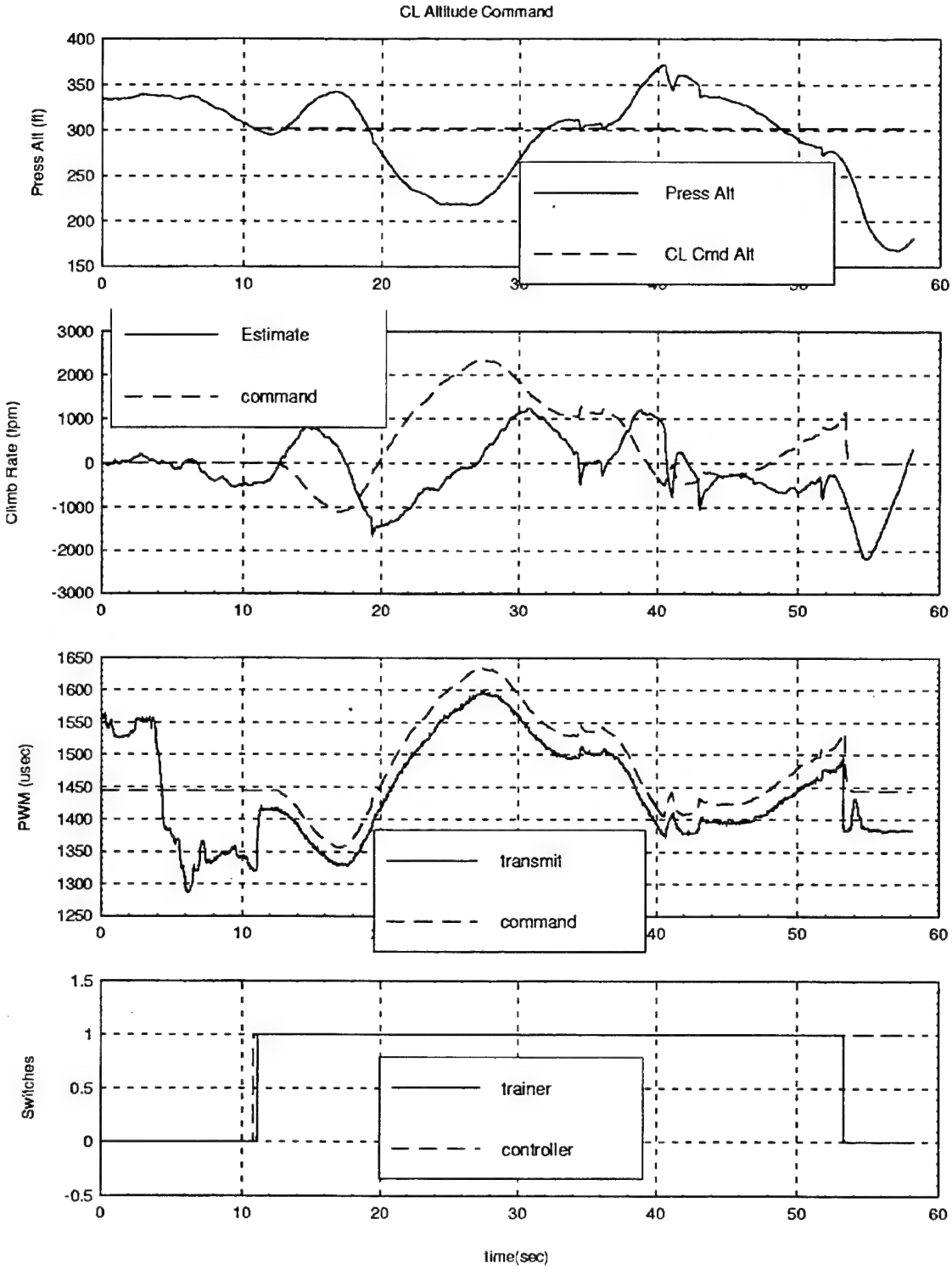


Figure 5.7: CL Altitude Command (Original Conversion Constants)

D. HEADING CONTROLLER

As mentioned earlier, close visual contact (less than ½ mile) was maintained with the FROG to facilitate accurate control and monitoring by the RC pilot. This made it impossible to test the tracking accuracy of the heading controller for any length of time without also maneuvering the aircraft with additional heading commands. Consequently, CL control with steady command inputs was very difficult. The OL tests were only conducted to verify the turn rate to PWM conversion formula and the PWM to volts calibration data. Therefore, the majority of the fight test runs was conducted exercising the CL heading controller. The results of the heading controller flight tests are highlighted below.

1. Open Loop Commands

The OL mode is of limited practical use since it can only command a specific turn rate through the FROG's autopilot. It cannot track an assigned heading, and without feedback can only command the correct yaw rate if the conversion formula constants and calibration data are correct. Flying the UAV is difficult in the OL mode, because of large time system delays (2 seconds) from command entry and slow roll response of the FROG using ailerons alone. Since there is no heading feedback or referencing, wind gusts will alter heading frequently. However, it is extremely valuable as an initial systems check before proceeding to CL controller tests. By engaging the OL mode with the Trainer switch and a zero turn rate command input, it is immediately obvious whether the software is "trimmed" correctly with the deg/sec to PWM conversion and the PWM to volts calibration.

Figure 5.8 shows an example of the OL mode when the controller is well calibrated. There is little or no change when open loop control is turned on with zero turn rate commanded. The data were recorded during data run three, flight one on 28 August 1998. The flight parameters were steady at 62 kts in a slow descent from 320 ft to 200 ft. The top strip chart shows a right turn prior to the Trainer switch coming on at 3.5 seconds

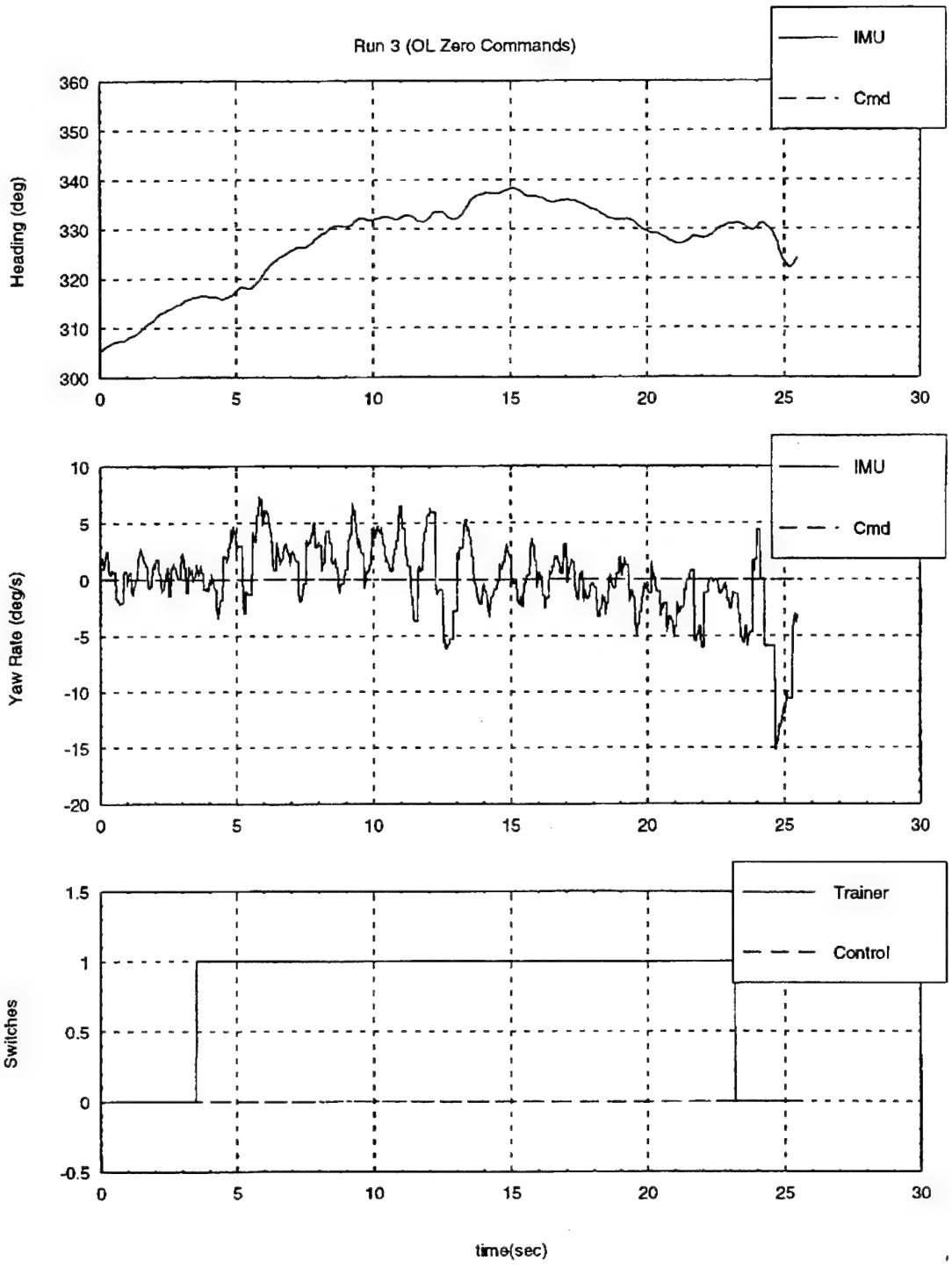


Figure 5.8: OL Heading Command

(bottom graph). The heading then stabilizes around 330°. The second graph shows the yaw rate oscillating ±3 deg/sec about the zero command line. The bottom graph confirms the Controller switch was OFF (zero value) and that the OL mode was active for 20 seconds before the pilot took back control due to altitude. Figure 5.9 confirms that the calibration from PWM to volts by the Ground Station is accurate. The plot shows the PW of the controller output is the same as the PW of the Futaba transmitted signal for turn rate, while the Trainer switch is on.

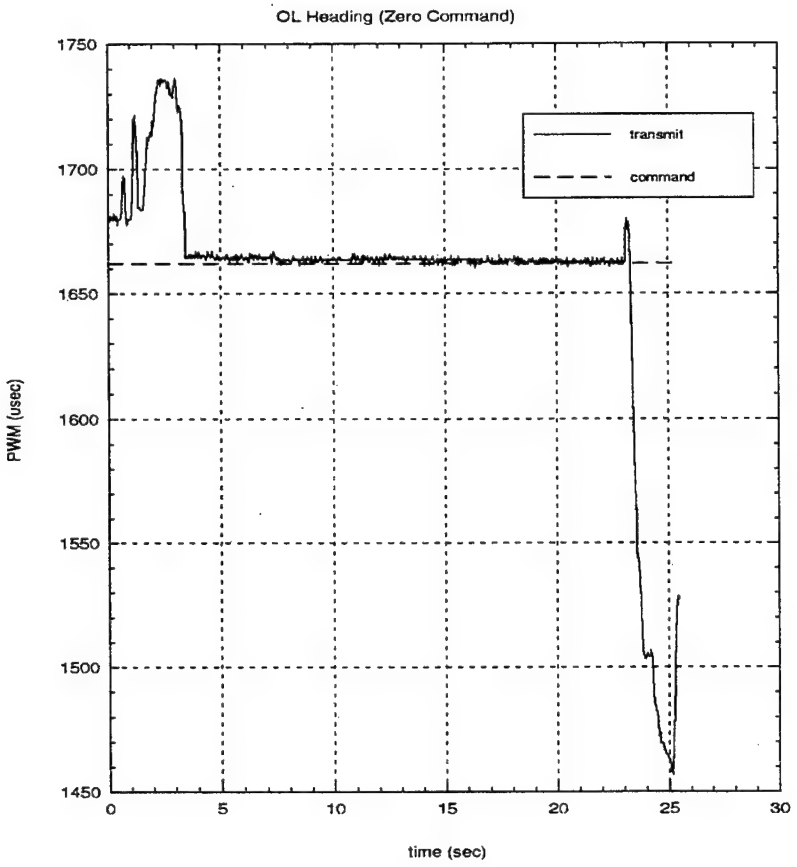


Figure 5.9: OL Heading Command Calibration

2. Closed Loop Commands

Complete evaluation of the CL heading controller was not possible due to the IMU heading problems discussed in Section A.3 of this Chapter. Sufficient data were

collected, however, to draw conclusions on the effectiveness and limitations of the controller as it is currently designed. CL heading control is demonstrated in Figure 5.10, where a 110° heading command is given followed by a 060° command. This data were recorded during flight two, data run 19 on 21 August 1998. The OL velocity controller maintained between 60 and 65 kts, and the CL altitude controller maintained 325 ft. The top graph shows the heading input (dashed line) plotted with the actual IMU heading. Recall the initial command of 360° is the entry code for zero command (heading hold). After the Trainer switch is activated, 110° heading is entered and as the aircraft reaches that heading a command of 060° is entered to keep the UAV within safe operating range of the pilot. The IMU heading starts at approximately 230° and follows the heading commands nicely. The second graph compares actual IMU turn rate (bottom trace) with what the controller output turn rate (top trace). The bottom graph confirms that both the Trainer and Controller switches were On for 54 seconds.

The robustness of the controller is evident by its ability to compensate for off calibration turn rate commands. Figure 5.11 shows data from the same closed-loop heading run that indicates an 80 μsec (5 deg/sec) difference has developed between the controller output (top trace) and the actual transmitted command (bottom trace). In other words, the controller wants a turn 5 deg/sec more to the right than the actual signal is commanding to the FROG autopilot. Another indication of robustness was the controller's ability to filter out heading input reversals from the IMU when approaching a North heading. This was demonstrated during several data runs, when the heading controller continued the turn in the correct direction until the IMU heading stabilized in the correct direction. Figure 5.12 shows one such example in the same format as Fig. 5.10. On this particular run, a right turn was commanded to heading 220° from 060°. While the IMU heading erroneously spun to the left through 360° and then reacquired the correct heading, the controller continued to command a positive (right) turn rate.

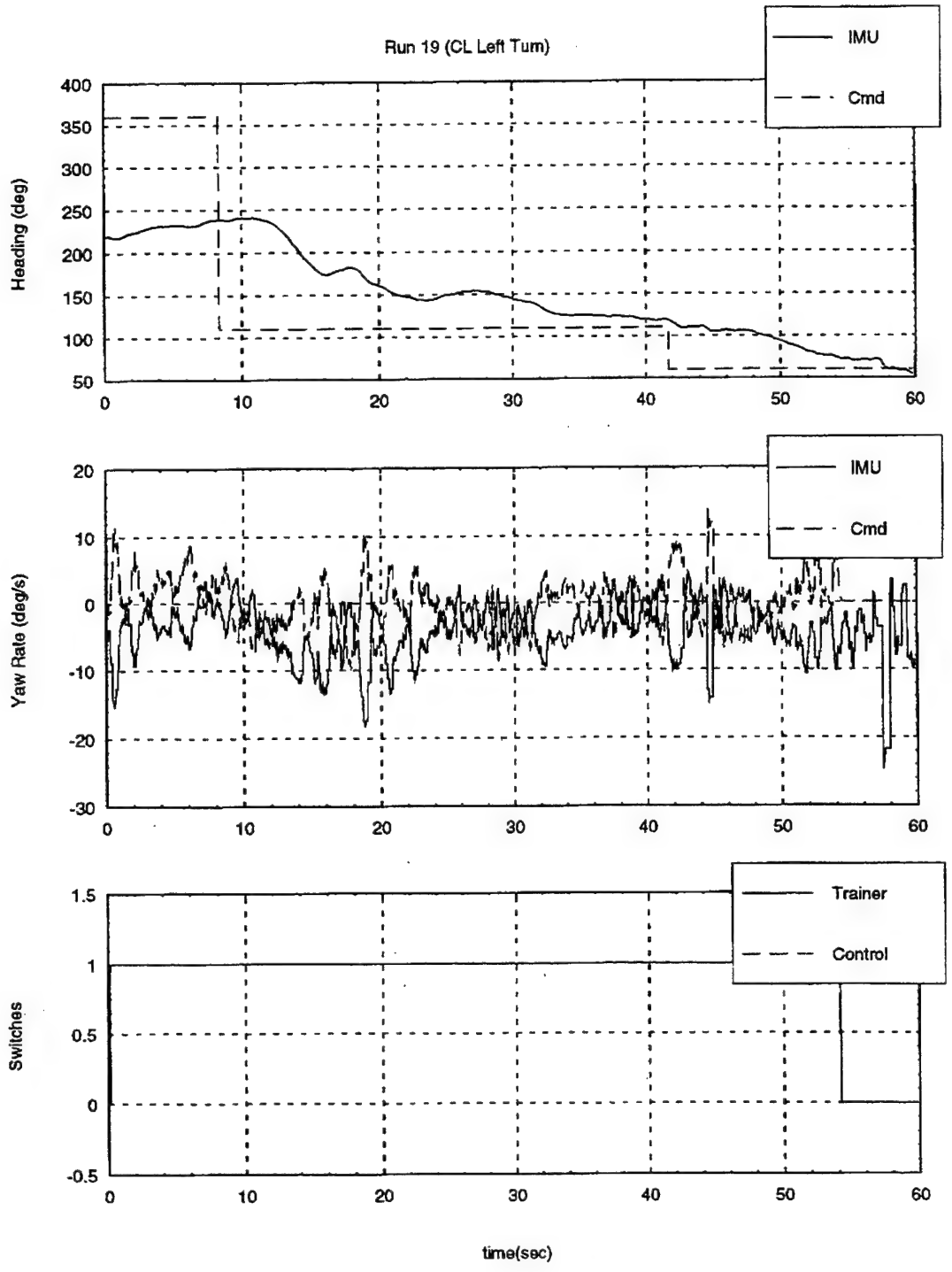


Figure 5.10: CL Heading Command With Calibration Error

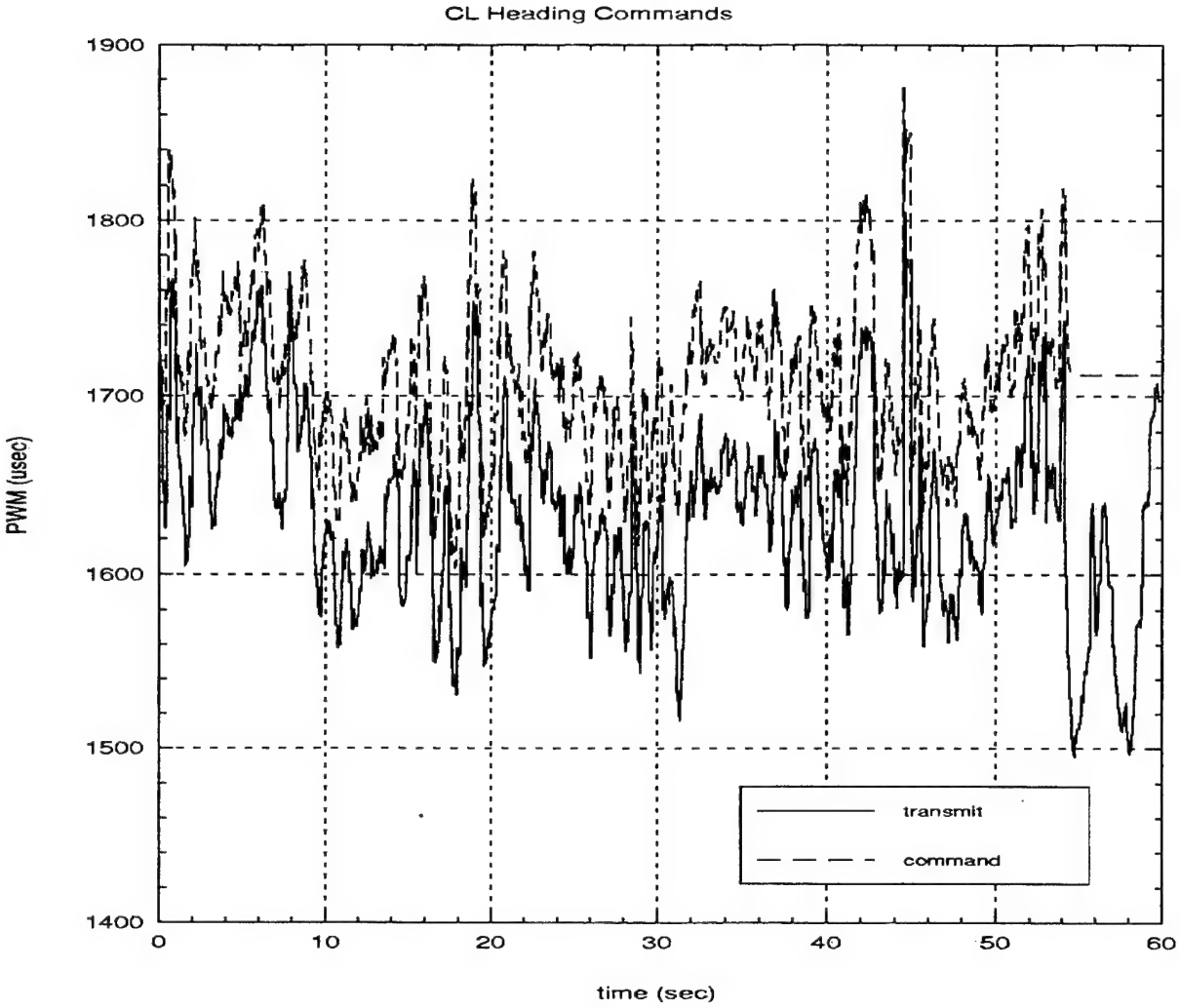


Figure 5.11: CL Heading Command Calibration

The current limits set in the heading controller integrator of ±20 deg/sec max turn command were never exceeded. However, significant altitude loss would result from turns greater than 10 to 15 deg/sec even if a small climb command was preset in the altitude controller.

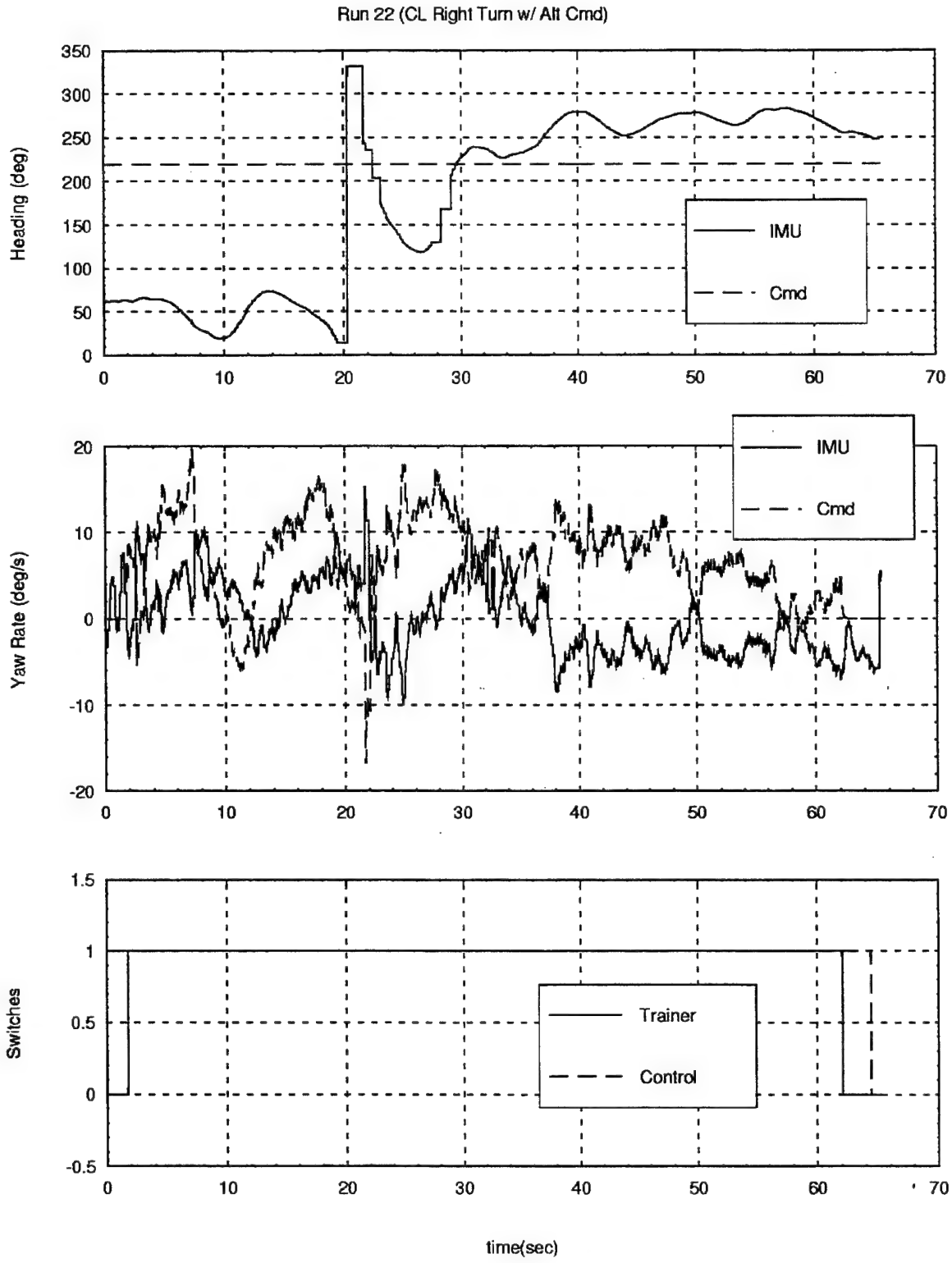


Figure 5.12: CL Heading Command With IMU Error

VI. CONCLUSIONS AND RECOMMENDATIONS

The two primary objectives mentioned in the introduction were accomplished:

1. All available sensors were evaluated for use with the FMS and the ones best suited were used.
2. A heading controller was designed, tested and implemented that met requirements and was robust in handling noise and temporary losses of heading information.

The RPS at the Naval Postgraduate School is extremely useful in designing, testing and implementing an FMS for unmanned air vehicles. The $MATRIX_X$ Product Family of software tools proved effective in building models of the hardware and controllers, as well as simulating their responses. Data acquisition and reduction, while time consuming, would have taken many more months without the tools offered by the RealSim programs.

The current FMS is fully functional and capable of safely controlling the airspeed, altitude and heading of the FROG; however, flight tests uncovered operational limitations, which should either be eliminated or safely accommodated before each of the controllers can be considered ready to support autonomous flight. Specific conclusions and recommendations follow.

A. SENSOR EVALUATION

The pitot-static system, due to its higher continuous data rate, was clearly a better source for airspeed and altitude than the GPS. Airspeed accuracy was judged similar for both sources, but the GPS altitude was less accurate and reliable than the static pressure altitude. Although noisier, the analog voltages from both pressure transducers could be filtered to acceptable levels for use by the controllers. The only advantage that could possibly justify using GPS airspeed is the need for accurate ground speed in navigation solutions.

The pitot airspeed indicated about –4 to –6 kts with the UAV stopped on the ground. If true airspeed accuracy is extremely critical for the mission, recalibration of the total pressure transducer is recommended with implementation of an operator selectable bias input similar to that which the pressure altimeter has. This would allow correcting for any constant errors that might develop in the system over time. The static pressure transducer is considered calibrated well enough for the FROG's current mission; however, the transducer has been relocated and the power supply changed since the last calibration was performed. Therefore, if altitude accuracy is critical, recalibration of the static pressure transducer is recommended also.

At the time of the flight tests, there was no acceptable source for heading data. The slow data rate and unreliability in maneuvers made the GPS an undesirable choice. The IMU's inability to indicate headings continuously during turns through North made it unacceptable for a heading control input. However, the IMU's design specifications indicate that given a velocity input it has the ability to estimate angle-of-bank and thereby improve the heading estimates. It is recommend that the IMU be reconfigured to accept velocity inputs from the pitot-static system and flight tests repeated for sensor and heading controller.

The sideslip indicator appears to be acceptable for use by a controller, but does have some noise present in the signal. Therefore, additional testing will be required once a sideslip controller is designed to determine what, if any, filtering is necessary.

B. AIRSPEED CONTROLLER

Once the conversion formula for PWM to knots was updated, the airspeed controller performed better in both OL and CL modes. Its response was quicker and tracking accuracy greatly improved. However, the low mass and lightly damped characteristics of the FROG UAV make it speed sensitive to gusts and maneuvers. Consequently, the closed-loop speed control task caused frequent and large amplitude

throttle changes, which were disconcerting to the pilot and in many cases unnecessary for safe completion of the maneuver.

Therefore, the OL speed control is the more practical mode and in fact more closely emulates actual pilot control technique. That is, a pilot of light aircraft will normally set the throttle for a desired speed and not change that setting to track minor airspeed changes caused by gusts or aircraft oscillations. It may be possible to adjust the controller gains to reduce the pilots concern for excessive throttle movements. However, this will also reduce responsiveness and degrade CL tracking accuracy. It will be very difficult to balance these requirements given the limited speed range and throttle control available to the FROG.

C. ALTITUDE CONTROLLER

With the addition of an accurate and reliable pressure altitude source, the altitude controller performance was better evaluated during flight tests. Its performance using the GPS altitude was erratic and difficult to analyze due to the large changes in indicated altitude from one GPS update to the next. The OL performance was improved by the updated conversion formula, which estimates a PWM value corresponding to the desired climb rate command. Specifically the new "x-intercept" value of 1340 resulted in little to no transient response when the OL mode was activated. Appendix C contains additional updated conversion formulas based on the latest flight test data.

Unfortunately, the new slope for the linear conversion formula resulted in the CL altitude controller going divergent in straight and level flight. Changing the slope effectively increased the controller gain on the output command beyond the gain margin. Data collected on the final day of flight testing was used to fine tune this conversion formula further. Recommend implementing and testing the latest slope value contained in Appendix C to determine if any instability still exists. Computer simulations should be run to reassess gain margins in the new configuration. Gain margins should be evaluated for both straight and level flight and turning flight, since the divergence only occurred in

the former condition. If the altitude controller diverges at all, recommend adjusting the gain and not the conversion constants to keep the controller stable.

Flight test data indicate that optimum gain values will differ significantly between wings level and turning flight conditions. Two recommended solutions are: either design a variable gain controller or limit the turn rate further.

Currently there are no limits placed on the climb rate command output of the altitude controller. Initially it was believed that the FROG autopilot had built-in limiters that would make this unnecessary. However, based on estimated vertical speeds achieved during the diverging altitude runs (±2000 fpm), it appears these limits are insufficient or nonexistent. Therefore, recommend limits of -500 to +1500 fpm be put on the output of the CL altitude controller for flight safety.

D. HEADING CONTROLLER

Once the conversion formula was updated the OL heading controller performed as expected, with no transients when activated and commanding turns in the intended direction. When the IMU was providing accurate heading data, the CL controller performed well also. Constant heading CL tracking errors were impossible to evaluate during flight testing due to the requirement for frequent turns to remain within a safe operating range as described earlier.

Because the current altitude controller cannot maintain altitude in turns greater than 10 to 15 deg/sec, it is recommended that the maximum output limits on the CL heading controller be further decreased to ±10 deg/sec.

E. SIDESLIP CONTROLLER

Although no design work was done on a sideslip controller, limited studies were conducted via computer simulation to evaluate the effectiveness of implementing a simple aileron-rudder interconnect (ARI) scheme to improve turn control and reduce Dutch Roll oscillations. Initial results indicate that very limited benefits can be gained

from an ARI (see Appendix E). The optimum blending of rudder command to turn rate command is –0.6 for the FROG/autopilot model used (note: the negative sign is required due to the traditional sign convention used for rudder deflection). Less than 0.6 in magnitude showed no noticeable improvements, while greater than 0.6 magnitude showed a tendency to diverge. The ARI reduced the heading overshoot slightly (5°) and decreased both the magnitude and frequency of oscillations slightly in roll rate, turn rate and bank angle. More noticeable, though, were the reductions in turn commands and aileron deflections required. This leads to the conclusion that if sideslip control is required, then direct rudder control will most likely be necessary using sideslip feedback. At this point data is inconclusive to whether the sideslip and yaw oscillations observed warrant any sideslip control. Sideslip sensor data and nose camera videotapes show that under most flight conditions the oscillations are minor or unnoticeable (±2°). However, several runs showed spikes as high as 10° due to gusts or maneuvering.

F. FLIGHT MANAGEMENT SYSTEM

The current FMS has improved in both design and function over the course of this project. Not only does it have an additional controller (heading), but also the performance of the original two controllers has been improved. The increased performance can be attributed to the addition of an altitude sensor and updated command signal conversion formulas. Design improvements include IA displays that are more user friendly. The benefits of both analog and digital displays were combined to improve operator situational awareness. A specific example of improving the operator interface is the addition of a "Master switch", which allows the operator to turn all three controllers on or off simultaneously. Before this required clicking on three different controller switches in different locations of the autopilot display. The controller switches have now been collocated with the Master switch.

It was clear that when properly calibrated, each controller worked well by itself. However, when using all three controllers simultaneously, it was observed that under

certain flight conditions one controller could command a maneuver that would be out of the other controller's limits. The most frequent example was the heading controller commanding a turn rate high enough to result in a bank angle, at which the altitude controller could no longer affect the climb rate. In the cases when the altitude controller began to diverge, the airspeed controller went from one extreme to another and eventually stayed at full throttle. Recommend considering two alternative solutions. The simplest approach might be to implement stricter maneuvering limits on each controller to ensure no one controller's capabilities are exceeded. However, the FMS may no longer meet mission requirements, if too many limits are imposed. Therefore, the second alternative is to consider blending the controllers' in such a way as to anticipate the coupling affects of the other controllers' commands.

Observing controller output data in various switch configurations during flight tests led to the suspicion that the Master switch had not been implemented in the same manner for all three controllers. Reviewing the block diagrams after the flight confirmed that the Master switch was not in the Heading Controller's Wind-Down Loop. Although the controllers' outputs were not used with the Master switch off, this omission permitted turn rate commands to build up while the Master switch was off and the Heading Controller switch was on. Recommend ensuring the Master switch is implemented in the same manner throughout the FMS. Another example of a difference in implementation is that the Master switch was included in controlling whether the airspeed commands come from an OL or CL source. However, the Master switch has no affect on the source of climb or turn rate commands.

Current procedures use PWM values at 2.4 volts and 2.7 volts output to the Futaba controller for calibration data. This calibration data is used to determine the linear relationship between a voltage signal into the Futaba controller and a PWM signal transmitted out of three different channels (elevator, aileron and throttle). It is uncertain exactly how linear these relationships actually are, but it is certain that if the linear approximation is to be valid at all, the calibration must be done across the same range of values expected in flight. Flight test data indicate that both the elevator and aileron

commands operate at voltages outside the current calibration range. Recommend the calibration voltages used be changed so that they more closely bracket the center or zero command value. The elevator voltage for zero climb rate is 2.05 volts. Therefore, recommend elevator calibration be performed at 1.9 and 2.3 volts. The aileron voltage for zero turn rate is approximately 2.76 volts. Consequently, recommend aileron calibration be performed at 2.5 and 3.0 volts (10 deg/sec is at 2.98 volts).

The data collected on the final day of flight tests has been added to the data used earlier to update the PWM-to-command conversion formulas. The recomputed conversion constants are included in Appendix C and should be used in future flight tests.

Ultimately, it was the intent of this thesis to further strengthen the foundation, on which to build a FMS tailored to specific mission requirements and able to effectively support autonomous flight of UAV's. Hopefully the data collected and analysis provided have done this.

APPENDIX A. SIDESLIP VANE CALIBRATION

A Spectrol Model 142 single-turn potentiometer was mounted in a specially manufactured plastic sleeve designed to slide onto the pitot tube and hold both angle-of-attack and sideslip sensors (see Fig. A.1). Figure A.2 shows a blown-up image of the potentiometer and Fig. A.3 contains its specifications and dimensions.

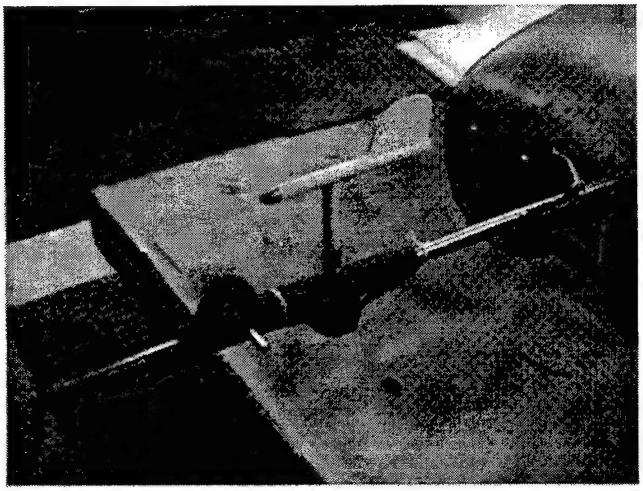


Figure A.1: Sideslip Vane

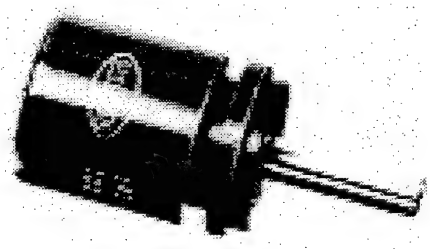
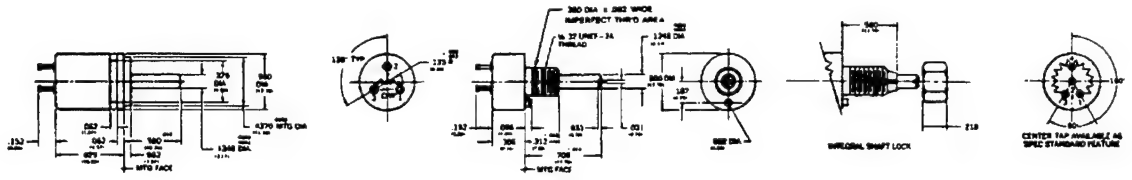


Figure A.2: Single-Turn Potentiometer Model 142



SERVO MOUNT

BUSHING MOUNT

TOLERANCES: UNLESS OTHERWISE NOTED,
DECIMALS ± .005, ANGLES ± 2°
BASIC DIMENSIONS ARE IN INCHES,
MILLIMETER DIMENSIONS IN PARENTHESES.

SCHEMATIC

HOW TO ORDER THE MODEL 140 OR 142

Models can be ordered from this specification sheet by stating:

140 - 0 - 0 - XXX

Model Number
140 - Bushing
142 - Servo

Mechanical Options
Bushing
0 - Stops, Slotted Shaft (std).
1 - Plain Shaft
2 - Shaft Lock
3 - Continuous Rotation
4 - Combination 1 & 2
5 - Combination 1 & 3
6 - Combination 2 & 3
7 - Combination 1, 2 & 3
Servo
0 - Continuous Rotation, Plain Shaft (std.)

Other Optional Features
Bushing
0 - Standard Torque
1 - Center Tap (10K max. Rt)
2 - High Torque
3 - Sealed Construction
4 - Combination 1 & 2
5 - Combination 1 & 3
6 - Combination 2 & 3
7 - Combination 1, 2 & 3
Servo
0 - Standard Torque
1 - Center Tap (10K max. Rt)

Resistance EIA Code

EIA Code	Standard Resistance	140 Resolution %	142 Resolution %
500	50Ω	0.542	0.542
101	100Ω	0.431	0.431
201	200Ω	0.361	0.361
501	500Ω	0.312	0.312
102	1KΩ	0.255	0.255
202	2KΩ	0.197	0.197
502	5KΩ	0.170	0.170
103	10KΩ	0.147	0.147
203	20KΩ	0.105	0.105

	140, 142	
Materials		
Housing	Aluminum, Anodized	
Bushing / Servo	Integral with Housing	
Lids: Rear	Thermoset Plastic	
Front	—	
Shaft	Stainless Steel	
Terminals	Brass, Gold Plated	
Bushing Mount Hardware:		
Lockwasher, Internal Tooth	Steel, Nickel Plated	
Panel Nut	Brass, Nickel Plated	
Electrical Specifications		
Dielectric Withstanding	1,000v RMS, 60Hz	
Insulation Resistance	1,000MΩ, 500v dc	
Peak Noise	100Ω ENR	
Environmental Specifications		
Moisture Resistant	Yes	
Vibration	20g, 10 to 2,000Hz	
Shock	50g	
Salt Spray	96 Hours	
Load Life	900 Hours	
Sealing	Available (M140 Only)	
Mechanical Specifications		
Bearing Type:		
Servo Mount	Ball	
Bushing Mount	Sleeve	
Operating Torque: oz-in	Starting	Running
Servo: 1 Section	0.075	0.05
Bushing: 1 Section	0.20	0.20
Each Add Section		
Mechanical Tolerance: in. Max	Bushing	Servo
Shaft Runout (TIR/in)	0.002	0.002
Pilot dia. Runout (TIR)	0.002	0.002
Lateral Runout (TIR)	0.003	0.002
Shaft End Play	0.006	0.004
Shaft Radial Play	0.003	0.002

Figure A.3: Single-Turn Potentiometer Specifications

In order to correlate the voltage output to degrees of sideslip, a special mechanism was designed by Don Meeks, the RC aircraft pilot. The instrument consisted of clamps and a protractor and fit on the pitot tube under the sideslip vane to provided a visual reference for measuring deflection angle of the sideslip vane (see Fig. A.4). The potentiometer output voltage was wired to one of the IMU analog input connections, where it is converted to digital format and transmitted via one of the RS-232 output channels to the Ground Station. An IA display was created to display the voltage value in digital format at the SPARC 2 workstation. The sideslip vane was rotated in both directions from -90° to +90° and voltage readings were manually recorded every 30°. Additional data points were recorded at ±10° and ±20°. The "polyfit" function of MATLAB was used to calculate a first order approximation between voltage and sideslip angle. As seen in Fig. A.5, the linear curve fit is accurate and the resultant formula is:

$$\beta = -39.5185(V) + 167.1632$$

The output of this formula is displayed both in analog and digital format on the Cal Air Data display page. In order to accommodate biases that may develop due to power supply changes to the potentiometer or inadvertent rotation of the potentiometer in the mounting, a slide switch was implemented on the same IA display. The Ground Station operator can use this switch to enter a constant value, which will be added to the above formula to correct for these biases. The easiest method to use this feature in the field without installing the calibration instrument is simply to align the sideslip vane with the pitot tube, take the displayed reading and enter that value times minus one. This will ensure an accurate zero reference.

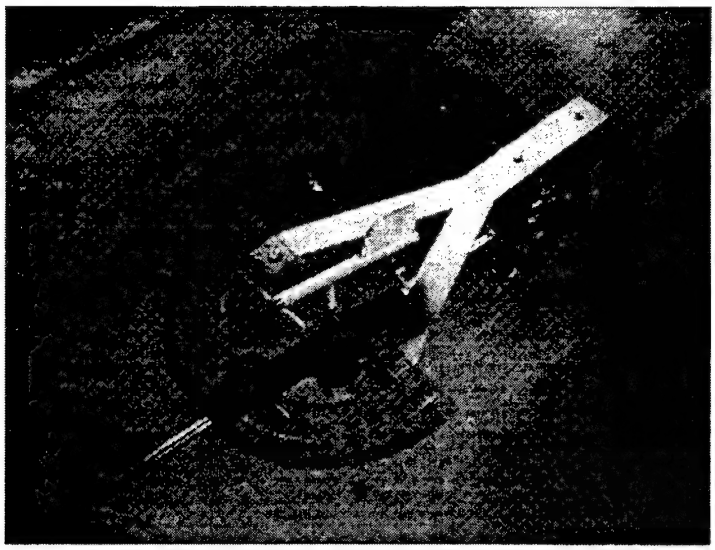


Figure A.4: Sideslip Calibration Instrument

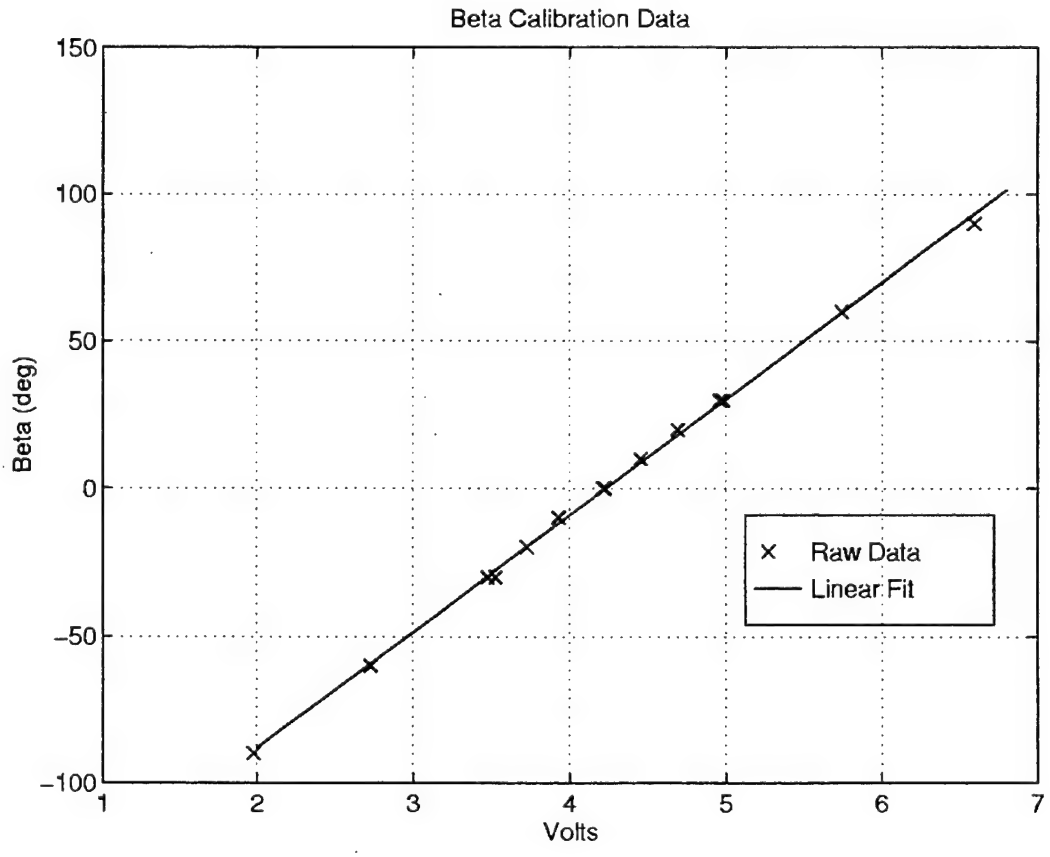


Figure A.5: Sideslip (β) Calibration Data

APPENDIX B. ALTIMETER CALIBRATION

A SensorTechnics barometric pressure transducer (Model 144SC1216BARO) was installed in the FROG to provide accurate pressure altitude data. A Schmidter, pictured in Fig. B.1, was used to apply a vacuum to the transducer and display the pressure changes in centimeters (cm) of water (H_2O). The pressure transducer output voltage was wired to one of the IMU analog input connections, where it is converted to digital format and transmitted via one of the RS-232 output channels to the Ground Station. An IA display was created to display the voltage value in digital format at the SPARC 2 workstation. The actual calibration voltage readings, however, were taken directly from the transducer output with a digital voltmeter to ensure accurate, noise-free readings. The vacuum pressure in the Schmidter was varied twice in both directions from 34 cm to 6 cm (31.2 cm being equal to atmospheric pressure) and voltage readings were manually recorded. A barometer was used to note current atmospheric pressure for conversion of cm of H_2O to pounds per square inch (psi).

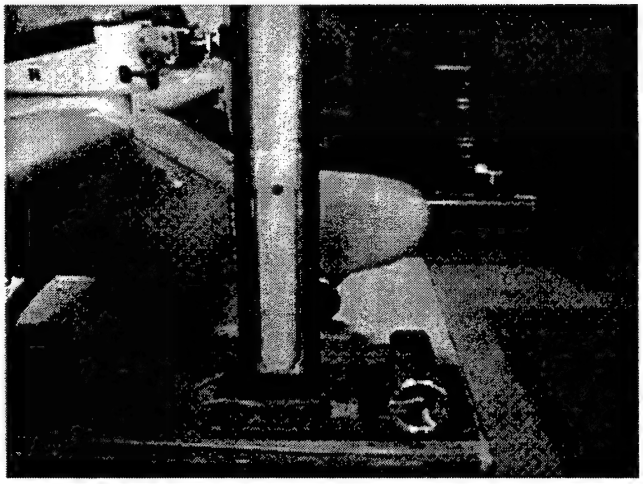


Figure B.1: Altimeter Calibration Equipment

The standard atmospheric formula:

$$\frac{P}{P_0} = [1 - 6.87535 \times 10^{-6}(h)]^{5.2561}$$

was used to convert psi to feet. The "polyfit" function of MATLAB was used to calculate a first order approximation between voltage and pressure altitude in feet. As seen in Fig. B.2, the linear curve fit is accurate and the resultant formula is:

$$h = -1519.1(V) + 5154.2$$

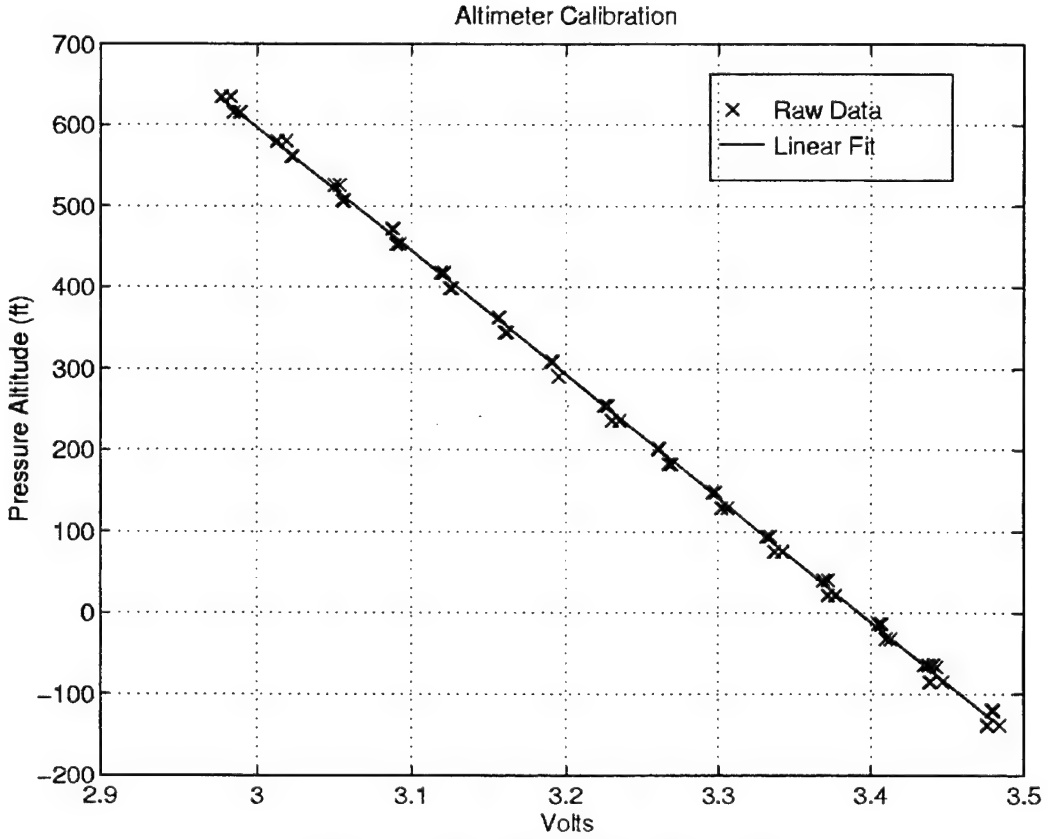


Figure B.2: Altimeter Calibration Data

The output of this formula is displayed both in analog and digital format on the Cal Air Data display page. In order to accommodate changes in local barometric pressure from Standard Day, a slider switch was implemented on the IA display in the same way as for sideslip bias correction. The Ground Station operator can use this switch to enter a constant value, which will be added to the above formula to correct for non-Standard Day pressures or field elevation. The easiest method to use this feature in the field without knowing field elevation or barometric pressure is to take the displayed reading before takeoff and enter that value times minus one. This will ensure an accurate zero reference for the ground.

As discovered during calibration checks, this sensor puts out extremely small voltage changes for altitude changes in feet (approximately one millivolt to 10 ft). Consequently, the instrument proved very sensitive to noise. Initial installation in the nose of the UAV resulted in noise actually being greater than the transducer signal. Therefore, the transducer was installed in the aft end of the fuselage as far away from electrical equipment and transmitters as possible. It was provided its own 9-volt battery power source to further isolate it from noise in the aircraft's power system. In addition, noise filters were added to the circuits and a software filter implemented in the Ground Station to further increase the signal to noise ratio.

APPENDIX C. CONTROLLER COMMAND CALIBRATION

Each controller output has to be converted to an equivalent PW value so that the PWM-to-volts calibration can then be used to convert it to equivalent volts. This analog voltage value can then be sent to the slave Futaba controller, where it is converted back to a PWM signal for transmission to the FROG. Without accurate conversion formulas, the OL controllers would be useless and the CL controllers seriously degraded. Therefore, five flight test data runs were dedicated to collecting data on the PWM-to-turn rate and PWM-to-airspeed relationships. In addition, all flight test data was examined for these relationships and the PWM-to-climb rate relationship. Only runs, where the given flight parameter was reasonably steady, were used to build a collection of data points. For example, in a turn, if the aileron command signal had a constant PW value and the IMU was showing oscillations about an easily identifiable mean turn rate, then the two values were paired together in the data base.

The "polyfit" function of MATLAB was used to calculate a first order approximation for each of the three controllers. Figures C.1 through C.3 show the results of these tests for the airspeed, altitude and heading controllers respectively. Each compares the latest linear fit with the raw data points and original fit being used in the FROG before the last day of flight tests. The conversion updates used on the last day of flying are only slightly different from the linear fit in these figures. The resulting formulas that should be used to update the Ground Station code are:

$$tp13 = \frac{V}{0.0338} + 53.3723$$

$$tp7 = \frac{Zdot}{7.0472} + 1322.4$$

$$tp8 = \frac{r}{0.0687} + 1664.2$$

The formulas have been put in the format currently used in the FMS, where tp13 is the label given the throttle signal, tp7 is the elevator command signal, and tp8 is the aileron signal.

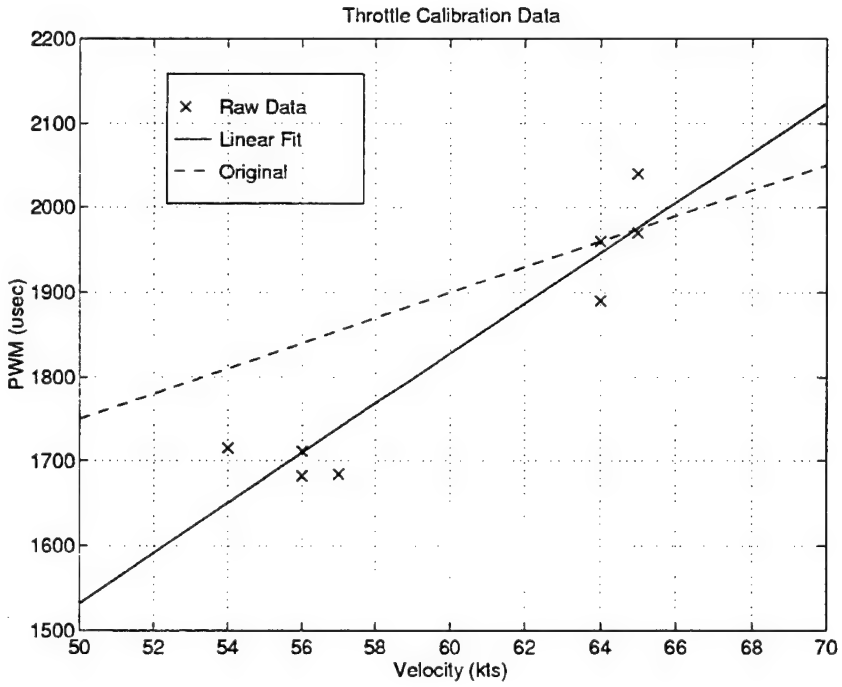


Figure C.1: Airspeed Command Calibration

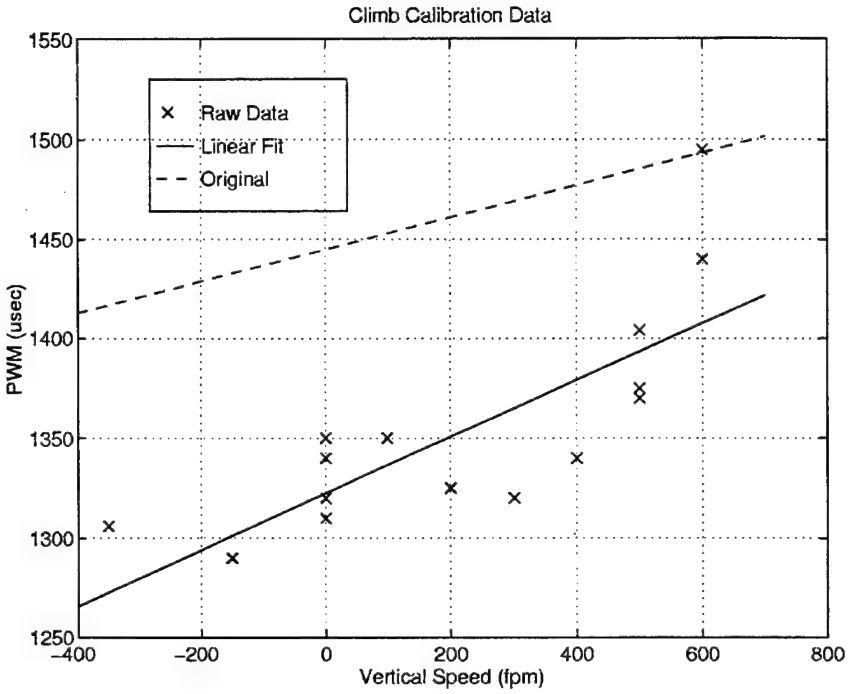


Figure C.2: Climb Rate Command Calibration

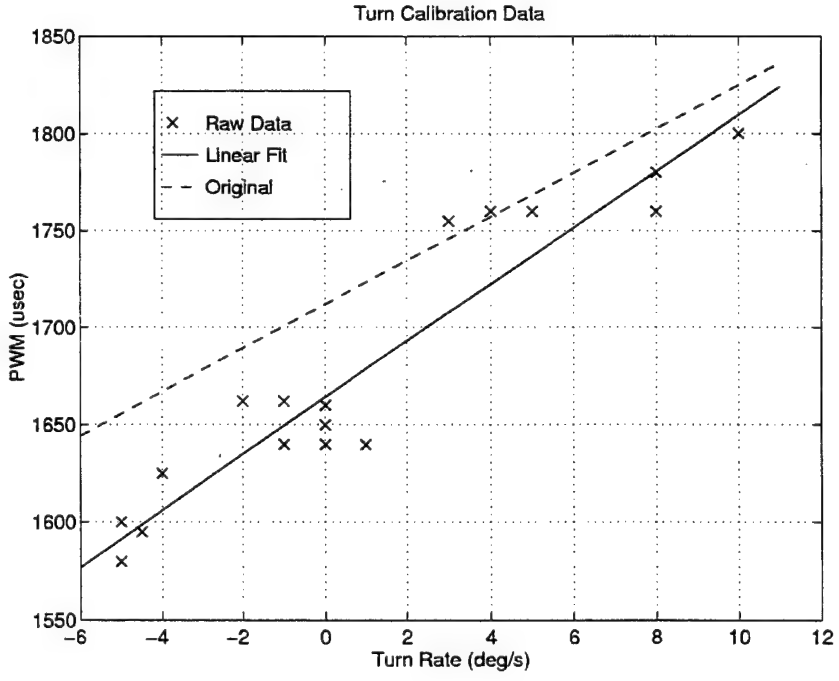


Figure C.3: Turn Rate Command Calibration

APPENDIX D. HEADING CONTROLLER USE OF MATHSCRIPT

The heading controller takes a desired heading command from the operator in a range of 0° to 360°, which is then compared to the heading input from the desired sensor. Since various sensors use different heading scales, a code was written to guarantee the headings being compared are in the same scale. The code was written in MathScript [Ref. 7], which is computer language used by Xmath and imbedded in a SuperBlock. Hence, the name "BlockScript" is seen in block diagrams where MathScript has been used. It is similar to high-level programming languages like C. Rather than list the MathScript code that may not be familiar to most, the code used to scale the heading input is expressed below using plain language.

Let u = heading input.

If $u < 0$, let $k = 1$.

Otherwise, let $k = -1$.

While the absolute value of $u > 360$, then replace u with $u + k\ (360)$.

When the absolute value of $u \leq 360$, then stop.

If $u < 0$, then replace u with $360 + u$.

If $u \geq 0$, output u as heading.

(This will work for any heading from $-\infty$ to $+\infty$)

Two other concerns for heading controllers is to ensure it always turns in the shortest direction to the commanded heading and how to hold heading, if "zero" is already a command heading. The following logic sequence is equivalent to the MathScript used to address these concerns:

Let u = heading commanded – actual heading from sensor (heading error).

Let x = heading commanded.

If x = 360, then replace u with zero and go to the end (output zero heading error).

If u < 0, then let k = 1.

Otherwise, let k = -1.

If the absolute value of u > 180, then replace u by u + k (360).

If the absolute value of u < 180, then output u.

(this code requires the heading inputs to be scaled properly by previous code)

APPENDIX E. ARI COMPUTER SIMULATION

The following figures are provided to support the conclusions and recommendations discussed in Chapter VI, Section E. Figure E.1 compares heading responses to a command requiring 180° heading change. Note the decrease in oscillations and overshoot using an ARI value of –0.6 (i.e., a command equal to –0.6 times the turn rate command is sent directly to the FROG model's rudder). Figure E.2 shows that the ARI only slightly reduces oscillations in roll, yaw and bank angle. Figure E.3 indicates that greater improvements are seen in the reduction of required commands and control surface deflections. Note, however, that the noise is greater and may result in control surface flutter with ARI on.

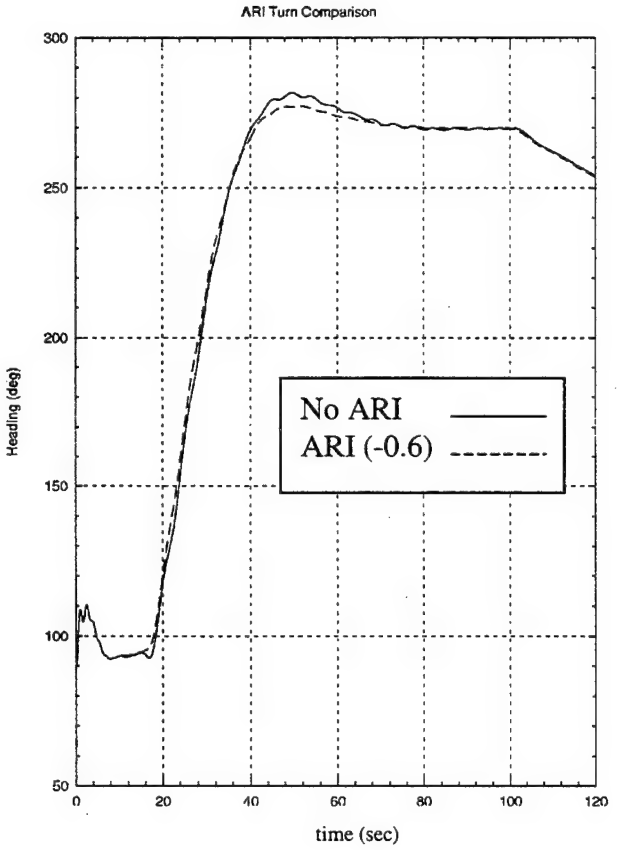


Figure E.1: ARI Heading Comparison

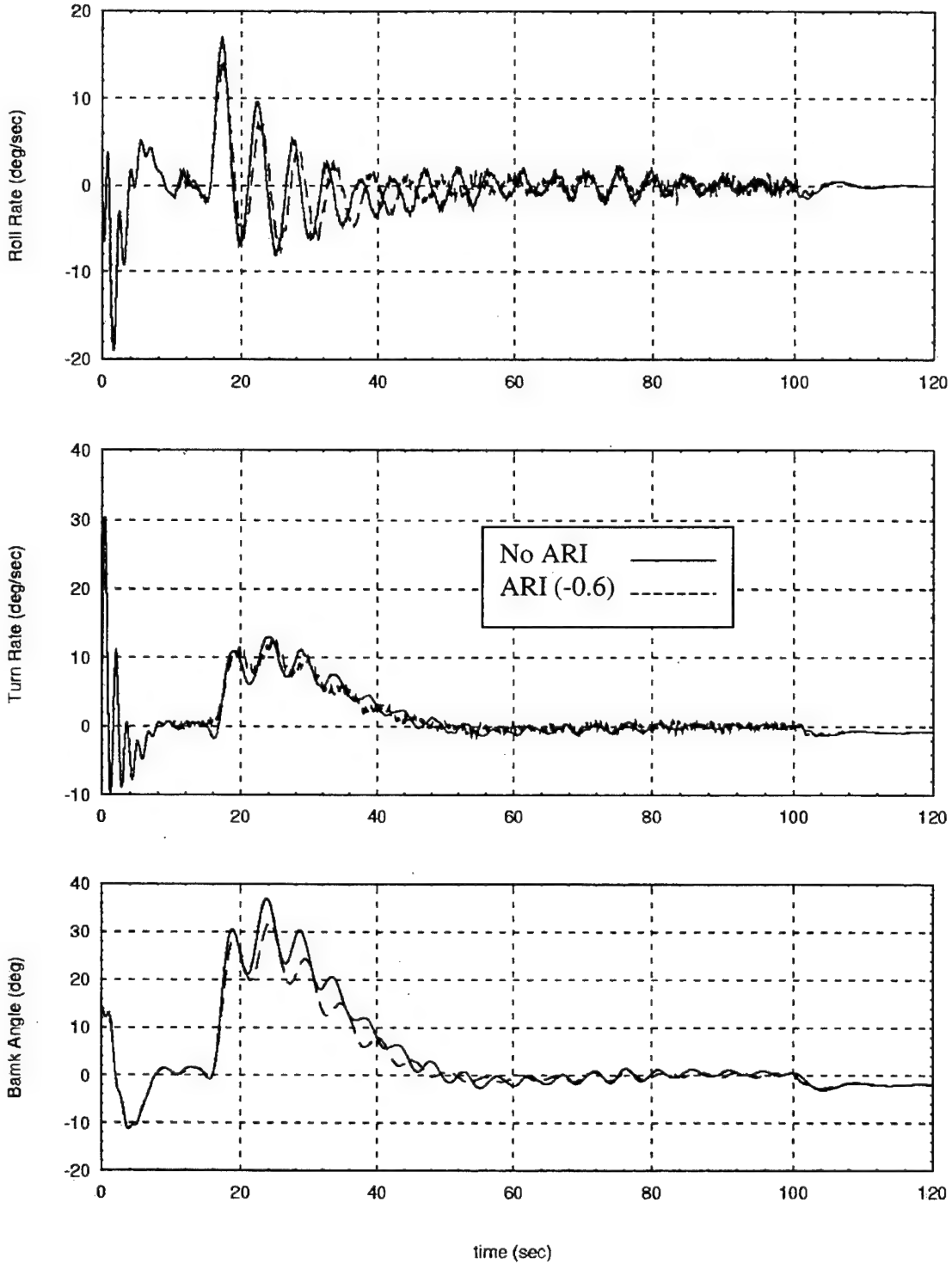


Figure E.2: ARI Performance Comparison

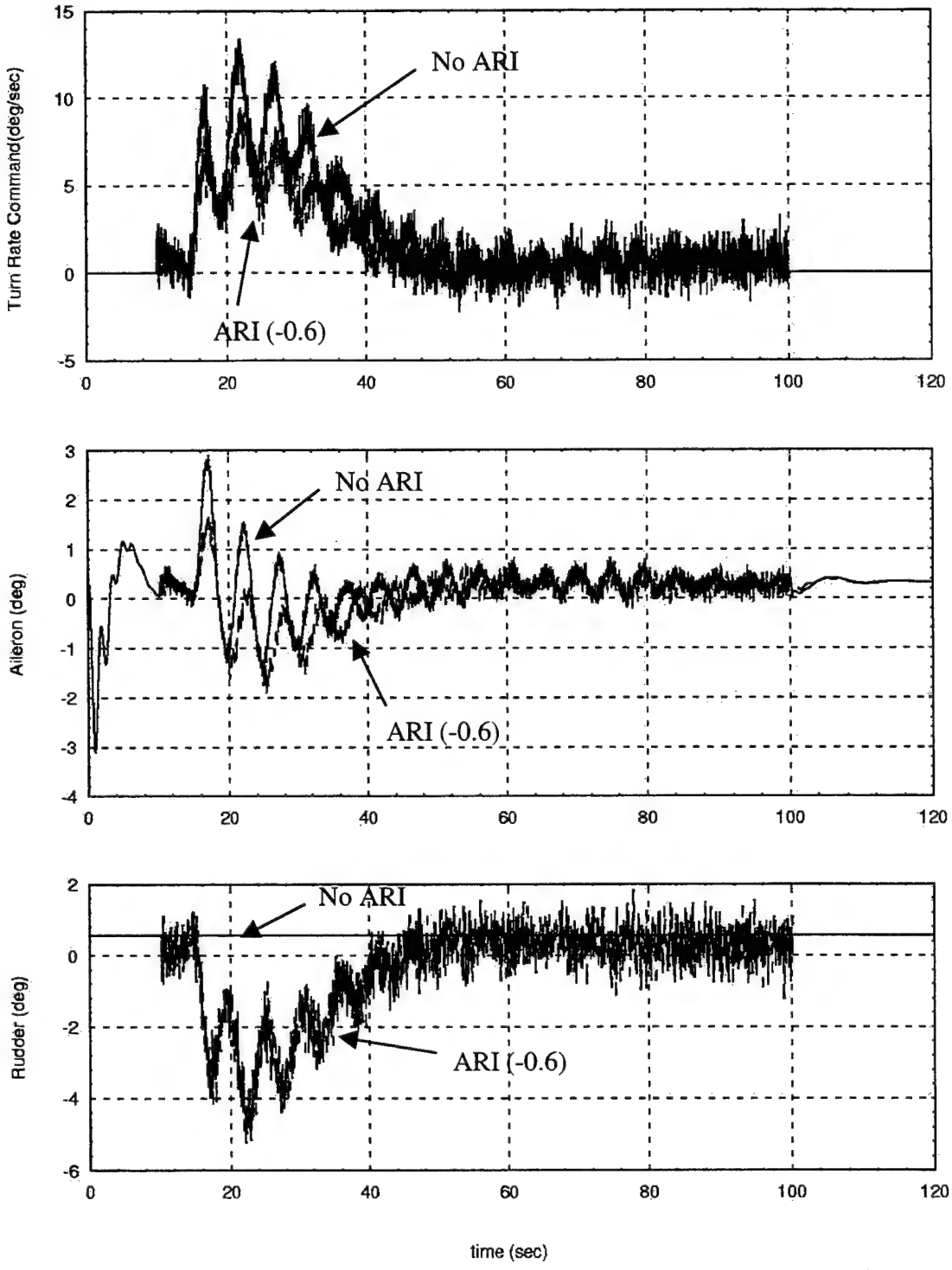


Figure E.3: ARI Command Comparison

LIST OF REFERENCES

1. National Defense Panel, Transforming Defense National Security in the 21st Century, p. 34, 1997.

2. Komlosy, J.A., *Applications of Rapid Prototyping to the Design and Testing of UAV Flight Control Systems*, Master's Thesis, Naval Postgraduate School, Monterey, CA, March 1998.

3. Froncillo, S.J., *Design of Digital Control Algorithms for Unmanned Air Vehicles*, Master's Thesis, Naval Postgraduate School, Monterey, CA, March 1998.

4. Hallberg, E.N., *On Integrated Plant, Control, and Guidance Design*, Ph.D. Dissertation, Naval Postgraduate School, Monterey, CA, September 1997.

5. Zanino, J. A., *Uniform System for the Rapid Prototyping and Testing of Controllers for Unmanned Aerial Vehicles*, Master's Thesis, Naval Postgraduate School, Monterey, CA, September 1996.

6. Allen, P.M., *Incorporation of a Differential Global Positioning System (DGPS) in the Control of an Unmanned Aerial Vehicle (UAV) for Precise Navigation in the Local Tangent Plane (LTP)*, Master's Thesis, Naval Postgraduate School, Monterey, CA, March 1997.

7. Integrated Systems Inc., *MATRIX$_x$ Product Family System Manuals,* Version 5.0, 1996.

8. FreeWave Technologies, Inc., *FreeWave™ Wireless Data Transceiver User Manual*, V3.4, 1996.

9. Watson Industries, Inc., *Inertial Measuring Unit Owner's Manual Part Number: IMU-BA604*, Revision F, May 1997.

10. Papageorgiou, E. C., *Development of a Dynamic Model for a UAV*, Masters Thesis, Naval Postgraduate School, Monterey, CA, March 1997.

INITIAL DISTRIBUTION LIST

No. of copies

1. Defense Technical Information Center.....2
   8725 John J. Kingman Rd., STE 0944
   Ft. Belvoir, Virginia 22060-6218

2. Dudley Knox Library.....2
   Naval Postgraduate School
   411 Dyer Rd.
   Monterey, California 93943-5101

3. Doctor Isaac I. Kaminer, Code AA/KA.....3
   Department of Aeronautics and Astronautics
   Naval Postgraduate School
   Monterey, California 93943-5121

4. Doctor Richard M. Howard, Code AA/Ho.....1
   Department of Aeronautics and Astronautics
   Naval Postgraduate School
   Monterey, California 93943-5121

5. Doctor Conrad F. Newberry, Code AA/Ne.....1
   Department of Aeronautics and Astronautics
   Naval Postgraduate School
   Monterey, California 93943-5121

6. Department of Aeronautics and Astronautics.....1
   Code AA
   Naval Postgraduate School
   699 Dyer Rd. Rm. 137
   Monterey, California 93943-5106

7. Commander Timothy C. Rivers.....3
   PEO(A), PMA-237
   Bldg. 2272, Rm. 154
   47123 Buse Rd.
   Patuxent River, MD 20670-1547

8. Avionics Lab, Halligan Hall.....1
Department of Aeronautics and Astronautics
Naval Postgraduate School
Monterey, California 93943-5121